University of Thessaly

Department of Mechanical Engineering

Diploma Thesis

Reliability and Maintainability Analysis in Semiconductor Manufacturing

by

Apostolos Stamatakis

Spyridon-Tsarls Botsis

Submitted in partial fulfilment of the requirements for the Mechanical Engineering Diploma from University of Thessaly.

Volos, October 2018

Approved by the Members of the Examination Committee:

First Examiner: Dr. George Liberopoulos
(Supervisor) Professor, Department of Mechanical Engineering,
University of Thessaly

Second Examiner: Dr. Dimitris Pantelis
Associate Professor, Department of Mechanical Engineering,
University of Thessaly

Third Examiner: Dr. Georgios K.D. Saharidis
Assistant Professor, Department of Mechanical Engineering,
University of Thessaly

Abstract

The motivation behind this thesis is the cooperation of our department, of Mechanical Engineering of the University of Thessaly, with the research project [redacted]. Through the project we were given three years' worth of data, from [redacted]'s semiconductor plant in [redacted], which our thesis is based on in parallel with the SEMI E10-0304E (revision of 2011), a standard specifically made for the semiconductor industry. We carry out a thoroughly statistical analysis for the UPTIME, PROCESSING TIME, PRODUCTIVE TIME, DOWNTIME, TIME TO RESTORE of each equipment and also compute sixteen metrics from the SEMI E10 metrics. We also have done a correlation research to find if there is any dependency between the pair of values PROCESSING TIME vs MAINTENCE TIME, PROCESSING TIME vs NUMBER OF MAINTENANCE, PROCESSING TIME vs TIME TO RESTORE, MAINTENANCE TIME vs TIME TO RESTORE as well as an autocorrelation for TIME BETWEEN FAILURES that may lead us to useful conclusions. Finally, we calculate the transition rate for each equipment. The mentioned procedures were automatically calculated with programs that were created by us.

Acknowledgments

We will like to thank our supervising professor George Liberopoulos for the opportunity he gave us to work on this diploma thesis and to work for a real-life case coming from the semiconductor industry. Also, for his guidance through all the time this thesis needed to be completed.

We also would like to thank his PHD student Michalis Deligiannis for his constant scientific support through all the procedure as well as his contribution for the on-time delivery of this thesis.

In addition, we will like to thank Mr. [redacted] from [redacted]'s [redacted] plant for the real-life data sample, the on-spot suggestions and all the help on the way which he provided without any hesitation.

Finally, but most importantly we would like to thank our families for all the constant support we had through all the years of our life to get where we are now and their help to build a strong foundation for our future.

Table of Contents

Table of Figures

Table of Figures in Appendix

List of Tables

1 Introduction

In this chapter we will analyze the objective our master thesis is based on and the problems we provide a solution for. We will continue with the Literature Review we did to learn and understand the problems and some approaches to the solutions in the semiconductor industry, then we will finish this chapter with the structure of the following chapters.

1.1 Thesis objectives

The objective of this thesis is to analyze the reliability and maintainability of the equipment used in the semiconductor industry. More specific we analyzed real three years' worth of data coming from the production line, of [redacted]'s semiconductor plant located in [redacted] in [redacted]. The plant manufactures chips and sensors for a large number of implementations, where many of them are for the automotive industry. The analyses are mostly based on the SEMI E10 standards, which is specifically made for equipment of the semiconductor industry. The SEMI E10 consist of a set of states the equipment must fall in and some metrics that help us determine the correct usage of the equipment.

The motives that made us work on this objective is firstly the desire to work on real-life data and to interact with a real and functioning plant instead with a theoretical problem or data. A big role also played the global trend to work and analyze big data for any implementation possible included the automation and optimization of the production in any plant that uses cutting edge technology. Last but not least is the importance of the semiconductors in the today's world and their number and complexity of the production procedures they follow, that creates the need for automation and optimization while making it a challenge. These reasons gave us the tenacity needed to overcome the obstacles we found during our research.

The opportunity to work on the real-life data and to work by a world leading company like [redacted] was given to us through the cooperation of our department of Mechanical Engineering, with professor George Liberopoulos in charge, with the research project «[redacted]». The goal of the project is to create a user platform across value chains and industries, thus promoting the digital networking of manufacturing companies, production machines and products [1]. It started the first of May 2017 and it will be active until the end of April 2020. The total budget of the project is 106 million euros coming from about 35% from government and European union funding and the rest from the private section, such as [redacted], BMW, Philips, [redacted] ABB, SAP, [redacted] VOLVO, [redacted] Karlsruhe Institute of Technology, Fraunhofer Gesellschaft, TU Dresden, Institute Mines-Télécom, TU Eindhoven, TU Wien, Politecnico di Milano, and many more.

1.2 Literature Review

The semiconductor industry is one of the fastest growing industries worldwide. The growth originates from the expanding scientific knowledge around the production of semiconductors as well as the improvement of the production management and the supply chain optimizations in the industry. In recent years, several researchers and practitioners have published on reliability analysis in semiconductor manufacturing.

Dhudshia (2006) presented software standards developed by the SEMI Information & Control Committee, whose charter is to explore, evaluate, discuss, and formulate consensus-based standard measurement methods, specifications, guidelines, and practices that, through voluntary compliance, will promote mutual understanding and improved communication between users and suppliers of manufacturing equipment and materials to enhance the manufacturing capability of the semiconductor and related industries.

de Jong et *al*. (2007) presented a simple and intuitive reliability qualification method that benefits from sub-system test cases. The proposed method was compared with the SEMI E10 standard, and two case studies showed the applicability for software reliability qualification.

Weidman (2009) showed several practical techniques used to gather reliability data to allocate efforts for improvements and presentation to customers. These techniques are based on basic reliability engineering concepts and were applied in simple ways. The presentation also reviewed definitions of several reliability engineering metrics.

Were et *al*. (2011) documented the revision SEMI E10-0304E, a Semiconductor Equipment & Materials International (SEMI) Specification for Definition and Measurement of Equipment Reliability, Availability, and Maintainability (RAM).

Munirathinam and Ramadoss (2014) explored predictive analytical algorithms and big data techniques to help achieve near-zero equipment downtime in the fab and to improve OEE (Overall Equipment Effectiveness), which is a key machine manufacturing productivity metric.

Weidman (2015) presented reliability and related metrics and explained how to move down a reliability requirement from the top-level system to the sub-systems and how reliability information is rolled up from components and subsystems to larger systems. He also mentioned some specific topics that contribute to reliability and described some general reliability types of analysis, such as a Failure Modes and Effects Analysis. The focus was electronics reliability.

The analysis presented herein is most closely related to Were et *al*. (2011) and the SEMI E10 standards, also borrowing some of the methodologies used in Liberopoulos and Tsarouhas (2004) who performed a statistical analysis of failure data of an automated pizza production line covering a period of four years. The analysis included the computation of the most important descriptive statistics of the failure data, the identification of the most important failures, the computation of the parameters of the theoretical distributions that best fit the failure data, and the investigation of the existence of autocorrelations and cross correlations in the failure data.

1.3 Thesis outline

In the next chapters we will show you the research we did for this dissertation. Their structure is explained in the following paragraphs.

The second chapter is aiming to inform about the production and appliances of the semiconductors. We will follow with a short explanation on the SEMI E10 standards, standards especially made for the semiconductor industry and a big part of our dissertation is based on them.

In the third chapter we will describe the three-year data sample given to us by [redacted]’s [redacted] plant. We will continue with the SEMI E10 and describe the data sample based on the SEMI E10. In addition, we give a short description of the equipment we used in our research.

In chapter four we will describe how we calculated sixteen metrics from the SEMI E10 standards, we will discuss the results of the equipment. After reviewing and compering the results with each other we will come to some conclusions.

In the fiive chapter we will explain how we worked using MATLAB to do a statistical analysis, we will inform you about the statistical distributions we used and why did we choose them. In the end we will show you a table with the results and also all the histograms with their closes fittings of distributions for uptime processing time, productive time, downtime and time to restore and the diagram for the hazard function of each fitted distribution.

In chapter six we will do a small correlation analysis using MATLAB. We will check for correlation for the same pair of values for each equipment and discuss the results.

In the seventh and final chapter we give you the calculated transfer rates matrix of each machine and a few conclusions on the results.

2 Semi-Conductor manufacturing

2.1 Semi-Conductor manufacturing

A semiconductor material has an electrical conductivity value falling between that of a conductor and an insulator. Their resistance decreases as their temperature increases, which is behaviour opposite to that of a metal. Usually the materials are used in semiconductor manufacturing are materials with solid crystal structure. Some examples of semiconductors are silicon, germanium, and gallium arsenide but also mixture of these. Silicon is a critical element for fabricating most electronic circuits. After silicon, gallium arsenide is the second most common semiconductor, used in laser diodes, solar cells, microwave frequency integrated circuits, and others. [2]

By themselves, intrinsic semiconductors are not of particular use. We can alter the properties of the material by introducing foreign substances or impurities into the crystal. These impurities are also known as dopants. A crystal with an added dopant is referred to as an extrinsic semiconductor or doped material [3].

Semiconductor applications are everywhere around in our life and they affect much of the world's industries. Those applications can be from a simple diode or transistor, to a microchip where its production is very complex and time consuming. [redacted]'s plant in [redacted] produces the later one and so as our thesis work on. So how is the semiconductor chip produced?

A semiconductor chip is a highly miniaturized, integrated electronic circuit consisting of thousands of components. Every semiconductor manufacturing process starts with raw wafers, thin discs made of silicon or gallium arsenide. Depending on the diameter of the wafer, up to several thousand identical chips can be made on each wafer by building up the electronic circuits layer by layer in a wafer fab. There are about 40 layers for the most advanced technologies. Next, the wafers are sent to sort or probe, where electrical tests identify the individual dies that are not likely to be good when packaged. Historically, bad dies were physically marked so that they would not be put in a package. Today, this has been replaced by producing an electronic map to identify the bad dies. The probed wafers are sent to an assembly facility where the dies with a reasonable quality are put into an appropriate package. Finally, the packaged dies are sent to a test facility where they are tested in order to ensure that only good products are sent to customers. Wafer fab and sort are often called front-end, and assembly and test are often called backend. While front-end operations are often performed in highly industrialized nations, back-end operations are typically carried out in countries where labour rates are cheaper. Considering the scale of integration, the type of chip, the type of package, and customer specifications, the whole manufacturing

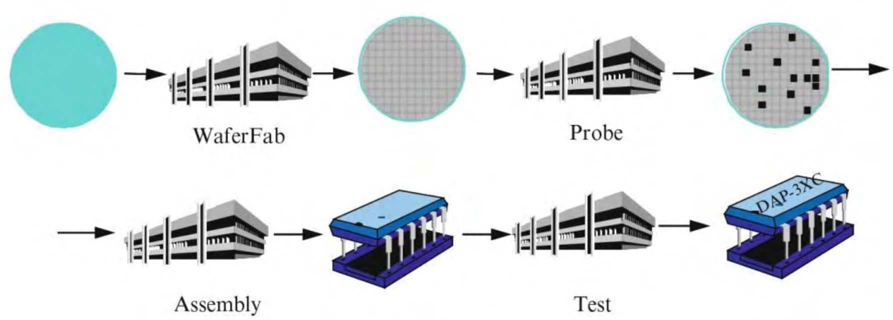


Figure 2.1The four main stages of semiconductor manufacturing

process may require up to 700 single process steps and up to 3 months to produce. The four main stages of semiconductor manufacturing are shown in Figure 2.1. In the past, all that was necessary for a semiconductor company to make money was to design a good product. However, over the last decade, increased competition has required semiconductor companies to also be able to manufacture their products in an efficient and cost-effective manner. Several performance measures are commonly used to describe and assess semiconductor manufacturing systems including machine utilization, production yield, throughput, cycle time, and on-time delivery performance-related measures. Machine utilization is extremely important because the machines account for around 70% of the cost of a new wafer fab, which can be as high as $5 billion USD. A high on-time delivery performance is important to satisfy customers. The competitiveness of a semiconductor manufacturer often depends on the ability to rapidly incorporate advanced technologies in electronic products, continuous improvement of manufacturing processes, and the capability of meeting customer due dates. In a situation where prices as well as the state of technology have settled at a certain level, the capability of meeting due dates along with the reduction of cycle time has become the most decisive factor in the fierce competition in the global market place. Consequently, short and predictable cycle times are highly desirable. Semiconductor companies have increasingly turned to data-intensive modelling and analysis tools and techniques because of their potential to significantly improve these performance measures, and hence the bottom line. The semiconductor manufacturing modelling and analysis community has been working over the last 20 years to modify general purpose manufacturing modelling tools and techniques to handle the intricacies and complexity of semiconductor manufacturing.

Description of the Base Process

Below is describing the base process steps of a wafer fab i.e., the operations, that can be performed in different work areas. The following process steps have to be performed in a wafer fab after starting the raw wafer.

1. Oxidation/diffusion: A layer of material is grown or deposited on the surface of a cleaned wafer. Oxidation aims at growing a dioxide layer on a wafer. Diffusion is a high temperature process that disperses material on the wafer surface. Diffusion furnaces and rapid thermal processing equipment are in place at the oxidation/diffusion work area. The furnaces are typical batch machines.

2. Film deposition: Deposition is used to deposit films onto wafers. The corresponding steps deposit dielectric or metal layers. There can be a dozen or more such deposition layers in an advanced circuit. Deposition can be executed by different processes, such as physical vapor deposition (PVD) or chemical vapor deposition (CVD), epitaxy, or metalization.

3. Photolithography: Coating, exposure, developing, and process control are the main steps of the photolithography process. In the first step, the wafer is coated with a thin film of a photosensitive polymer, called photoresist strip. Accurate and precise three-dimensional patterns are produced on the silicon wafer’s surface when an IC pattern is transferred via a photo mask, i.e., reticle, onto the photosensitive polymer, which replicates the pattern in the underlying layer. Exposure tools, called steppers, transfer the pattern onto the wafer by projecting light through the reticle to expose the wafer using ultraviolet light. The exposed wafer is then developed by removing polymerized sections of photoresist from the wafer. Every wafer passes through the photolithography area up to 40 times because the circuits are made up of layers. The photolithography work area is a typical example of a bottleneck in a wafer fab because steppers are very expensive machines.

4. Etch: This step is responsible for removing material from the wafer surface. The wafers are partially covered by photoresist strip after the photolithography step. Areas on the wafer that are not covered are then removed from the wafer. We differentiate between wet and dry etching. In the first case, liquids are used, whereas gases are necessary for the latter case.

5. Ion implantation: Dopant ions are selectively deposited on the surface of the wafer. Doping material is deposited where parts of the wafer have been etched. Ion implanters are used for between four and eight applications for most modern ICs.

6. Planarization: This step cleans and levels the wafer surface. It is called chemical-mechanical polishing (CMP). A chemical slurry is applied to a wafer and the surface is equalized. This results in the thickness of the wafers being diminished before adding a new layer. [4]

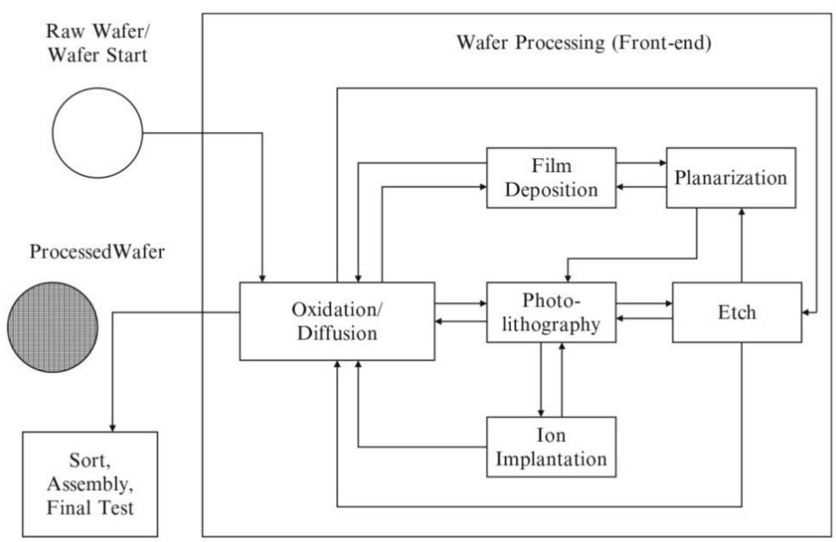


Figure 2.2 The base process steps of a wafer fab.

For all those equipment used in the above procedures there is a MRP system that tracks all the steps for the products but also all the state changes for the equipment. All of the information is stored in a Database in many different tables to later be analysed.

2.2 SEMI E10

SEMI E10 is a Semiconductor Equipment & Materials International (SEMI) Specification for Definition and Measurement of Equipment Reliability, Availability, and Maintainability (RAM).

The semiconductor industry has been instrumental in developing a methodology for tracking and evaluating the application of information from equipment regarding its operating condition (Trybula and Pratt 1994). Originally published in 1986 and revised in 1990, 1992, 1996, 1999, 2001, 2004, 2011 (SEMI E10-0304E) which we used in our analysis. SEMI E10 is the “father” of SEMI equipment performance and productivity metrics and is one of the most widely used SEMI standards. SEMI E10 establishes definitions for basic equipment conditions, RAM metrics, and equipment utilization measurement providing a common language and methodology between equipment suppliers and users. SEMI E10 has become a standard for tracking performance of the semiconductor manufacturing equipment and a benchmark for other industries. According to the Semi E10 specification the considered equipment's total time in organised as is described in the figure below.

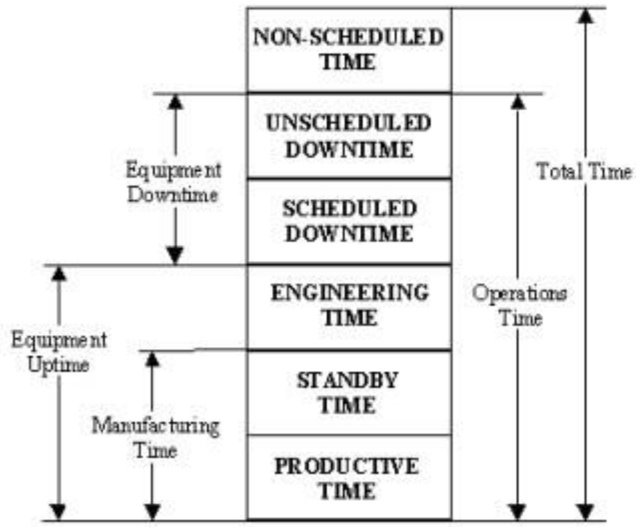


Figure 2.3 The six states from the SEMI E10 standards

SEMI E10 further defines equipment RAM and utilization metrics that provide the uniform common language for equipment users and suppliers to analyse equipment performance and identify improvement opportunities throughout the equipment life cycle. SEMI E10 supports supplier-user relationships by providing a basis to establish equipment performance requirements during purchase and service negotiations. The acceptance of SEMI E10 as the foundation for performance specification improves user-supplier relationships that stimulate cooperation and partnership and promotes continuous improvement in equipment design and support (Dhudshia 2008). [5]

3 Description of the data and data processing

3.1 Description of data

The analysis is based on the data sample that was released from [redacted] plant in the middle of July. The data sample contains seven kind of tables that come in the form of a txt file.

1. [redacted]
2. [redacted]
3. [redacted]
4. [redacted]
5. [redacted]
6. [redacted]
7. [redacted]

In all the tables, the products, equipment groups, and production areas are anonymized. For our analysis we will only need two kind of tables, the [redacted] tables and [redacted] table.

3.1.1 [redacted]

The [redacted] tables come in thirteen txt files and contain three years' worth of information. The Figure 3.1, below, shows a screenshot of one of the files.

[redacted]

Figure 3.1 The txt file of the [redacted] *table.*

The tables originate from the [redacted] of the [redacted] plant. The lines in the database get automatically written down but the MRP system of the plant. Every step of the value adding process chain of a lot is recorded at the entree and exit of each equipment it visits. The database table consist of many columns where only fifteen have been shared with us. The columns are:

[redacted]

Due to the big amount of raw data (more than fifty million lines of input) and files, the tables were transferred to a single table in a SQL database. The SQL database makes the tables easily accessible and can be easily configurated to show the information needed. As you can see below in Figure 3.2.

[redacted]

Figure 3.2 The [redacted] *seen from the SQL Database.*

3.1.2 [redacted]

The [redacted] table comes in one txt file and it also contains three years' worth of information. As you can see below Figure 3.3.

[redacted]

Figure 3.3 The txt file of the [redacted] table.

This table also originates from the [redacted] and is automatically updated from the same MRP system. The lines in the table recor a c ange n t e equipment state. The table contains the following columns.

[redacted]

The following table describes the meaning of each state.

States	Meaning
Null	If in [redacted] and [redacted] in {0,1} new equipment and the [redacted] is the start of production Else Unscheduled downtime
[redacted]	Available contains idle or processing
[redacted]	Available, idle
[redacted]	Maintenance
[redacted]	Short maintenance (visual check, a machine parameter adjustment, etc.)
[redacted]	Testing new Product or new parameter set (planned test run)
[redacted]	Machine needs to be seen by an engineer
[redacted]	Unscheduled down / Machine unplanned broken
[redacted]	Support material needed
[redacted]	Out of production

Table 1 Descriptions of States

This table was also added to the SQL Database for the same reasons as the previous table (see Figure 3.4 below).

[redacted]

Figure 3.4 The [redacted] *table seen from SQL Database.*

3.2 [redacted] states allocated to E10 States

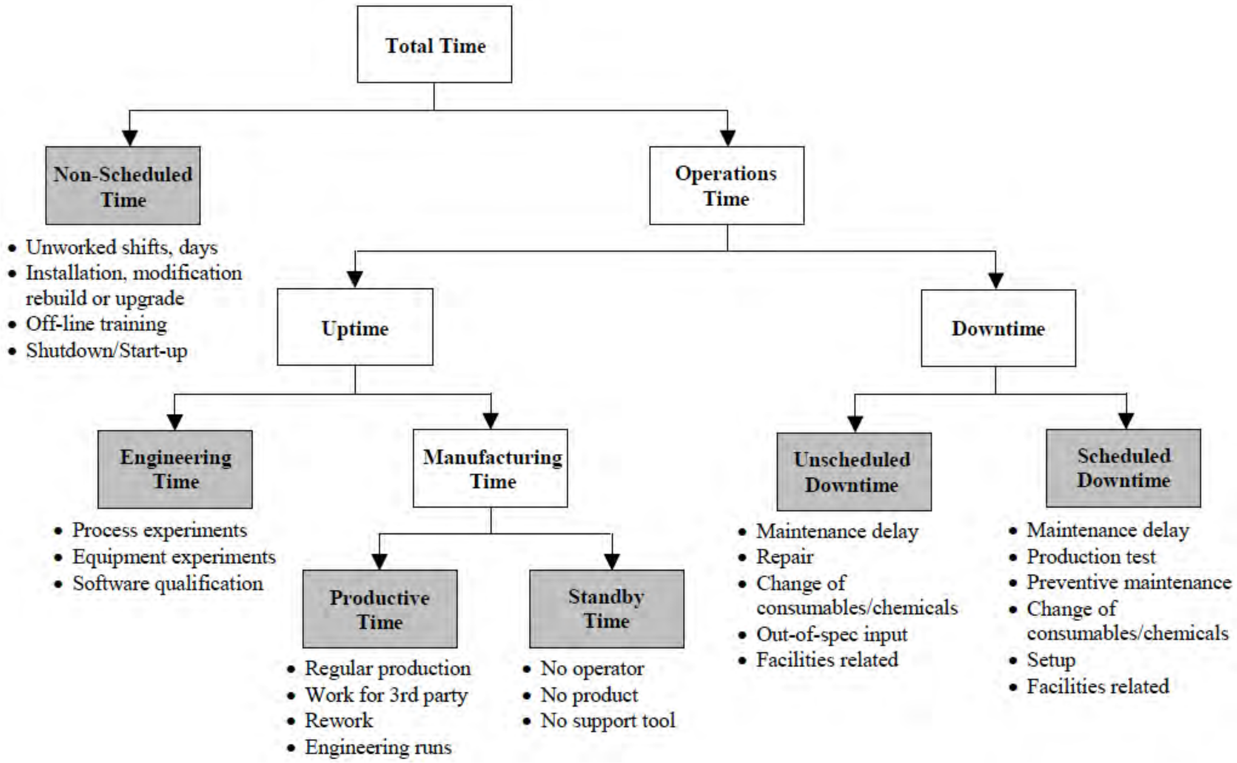


Figure 3.5Equipment State Hierarchy

The SEMI E10 protocol to measure equipment performance defines six equipment states into which all equipment conditions and periods of time must fall. Any equipment system must be in one and only one E10 state at any point in time. Any equipment system may only be subject to at most one system-level failure at any point in time regardless of the number of underlying constituent problems contributing to or arising from that failure.

For our analysis we have allocated the ten [redacted] states in the six equivalent SEMI E10 equipment states as seen in Figure 3.6.

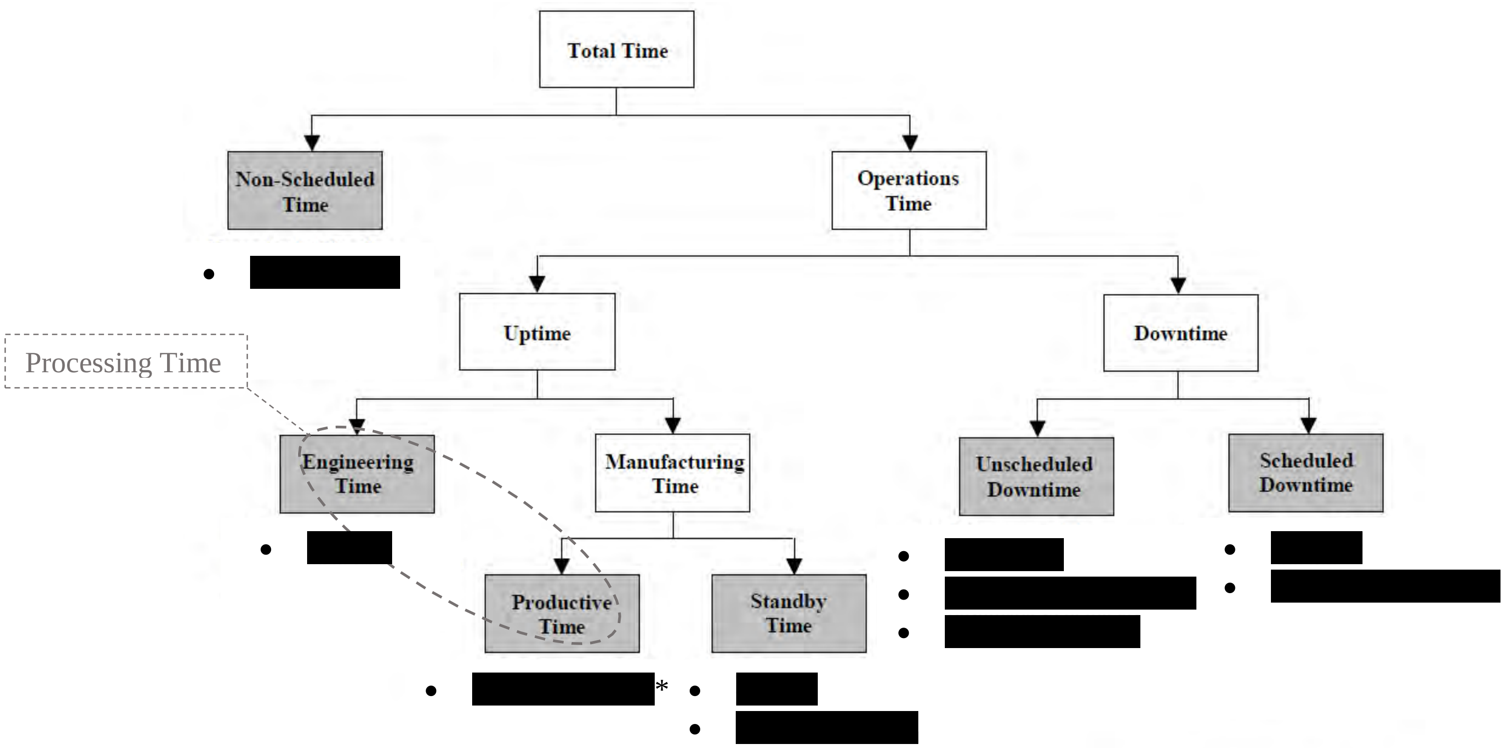


Figure 3.6 Equipment State Hierarchy for [redacted] equipment states with Processing time

- Null falls in unscheduled downtime if it does not mean the start of production of the equipment. This state is rarely found in the database and did not occur on the equipment we studied.
- The [redacted] state contains productive time and standby time. With the help from the [redacted] table we can determine whether the state falls in productive time or standby time.
- The [redacted] state falls in standby time
- The states [redacted] and [redacted] fall in the scheduled downtime as both are maintenance procedures.
- [redacted] falls in engineering time as the machine is working properly although for planned test runs.
- The states [redacted], [redacted] and [redacted] all fall in the unscheduled downtime and [redacted] is the only state that is equipment-related unscheduled downtime.
- The [redacted] state fits in the non-scheduled time as it is removed from the production network and there is no scheduled work for the equipment.

3.3 Data processing

From the [redacted] table in our SQL database we pull the information we need using SQL statements. We write a query to fetch lines in the database where we demand inputs for one equipment, the [redacted] is not equal to [redacted] (signaling that a state change has occurred) and all the lines should be ordered by [redacted] (ascending), we do this for each of the equipment we worked on and save as a excel file. We also follow a similar procedure for the [redacted]

[redacted] table. We demand the inputs for the equipment we are interested in and without the inputs that have [redacted] equal to [redacted] (zero process time). The lines are exported to an excel file ordered by [redacted].

Now for each of the equipment we have a set of two tables, and for our next step is to tackle the problem which the state [redacted] that falls in two states from SEMI E10. We do that by creating a program in Visual Basic that recognizes when the equipment is idle or productive from the [redacted] table and creates a table similar to the [redacted]. We assume the equipment is productive when there is at least one lot in the equipment, without taking in thought if the equipment is fully utilized (i.e., if it can process more than one lot) or not. We combine the newly created table with the other table and we create a table that has all the state changes needed for the SEMI E10 standards.

Having the combined table, we procced to find the time the equipment passes in each state for every unscheduled downtime cycle. An unscheduled downtime cycle is the time period between failures, where failure is any unplanned or unscheduled downtime event that changes an equipment system to a condition where it cannot perform its intended function. It is also a downtime event that changes the equipment system state to unscheduled downtime. One or more component or subsystem failures, software or process recipe problems, facility or utility supply malfunctions, or human errors could cause an equipment system failure. Subsequent problems occurring during the continuous unscheduled downtime are not counted as additional failures [6]. That also means that from the three years of information we are given we will compute the time for each state for the time between the first and last failure in those years.

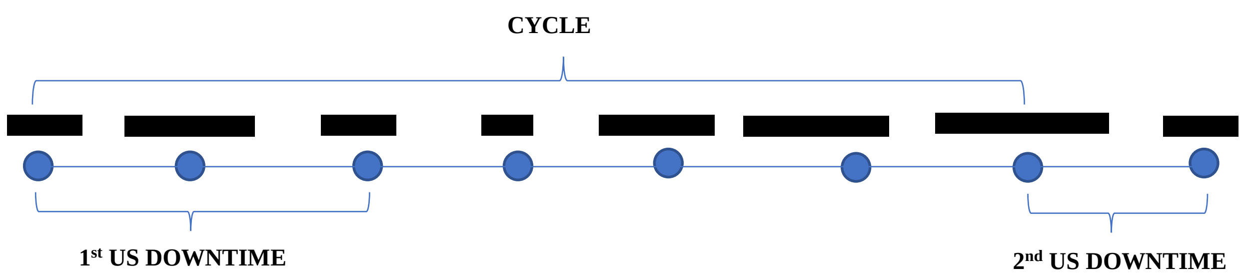


Figure 3.7 Unscheduled downtime cycle.

Having defined an unscheduled downtime cycle above, we proceed to define the times that the equipment spends in different possible states during such a cycle. These times are:

- uptime = sum of times that the equipment is in states [redacted], [redacted], and [redacted].
- processing time = sum of times that the equipment is in states [redacted] and [redacted]-productive.
- productive time = sum of times that equipment is ins states [redacted]-productive.
- downtime = sum of times that equipment is in states [redacted], [redacted], [redacted], [redacted], and [redacted].
- time to restore = sum of times that equipment is in all the states of unscheduled downtime ([redacted], [redacted], and [redacted]).
- maintenance time = sum of times that equipment is in all the states of scheduled downtime ([redacted] and [redacted])
- # of maintenances= number of instances that equipment is in all the states of scheduled downtime (counting once for consecutive maintenance states).

In the end we redo the same work with the difference that our cycles are between equipment-related failures. In our case those failures will be the unscheduled state [redacted]. Again, failures that occur continuously will be treated as one and time to restore will be from a [redacted] till we leave the unscheduled state. The other unscheduled downtime states are counted normally in downtime although do not affect the length of the cycle.

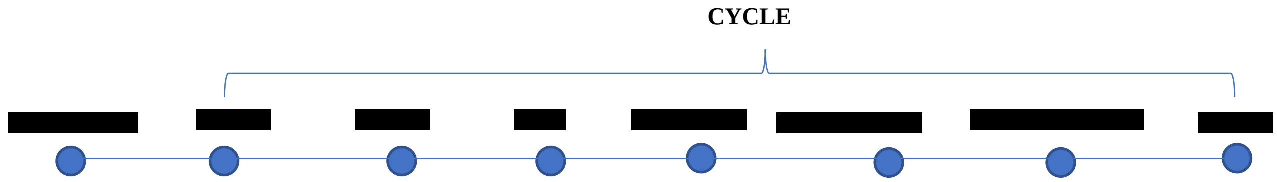


Figure 3.8 Equipment-related unscheduled downtime cycle.

All of the results are calculated in hours and for both of the cases the cycles with very small duration of uptime (<6 minutes) were simply merged with the next cycle.

We also calculated the following quantities needed for the computation of important metrics. Those are:

- **Uptime:** the total time the equipment was up between the first and last failure.
- **# of failures during uptime:** the number of failures where between the failures the uptime was not equal to zero. In this analysis it is equal to number of unscheduled life cycles because we do not accept zero uptime between failures.
- **Unscheduled downtime for failures during uptime:** the total time the equipment was unscheduled downtime for each cycle where uptime is not equal to zero. For this analysis it is equal to the total downtime due to the fact that we do not accept zero uptime between failures.
- **# of equipment-related failures during uptime:** the number of equipment-related failures where between the failures the uptime was not equal to zero. In this analysis it is equal to number of equipment-related unscheduled life cycles because we do not accept zero uptime between failures.
- **Productive time:** the total time the equipment was productive between the first and last failure.
- **# failures during productive time:** the number of failures where between the failures the productive time was not equal to zero. Is equal to the number of unscheduled life cycles where productive is not equal to zero.
- **Unscheduled downtime during productive time:** the total time the equipment was unscheduled downtime for each cycle where productive time is not equal to zero.
- **# equipment-related failures during productive time:** the number of equipment-related failures where between the failures the productive time was not equal to zero. Is equal to the number of equipment-related unscheduled life cycles where productive is not equal to zero.
- **Total time:** the time between the first and the last failure.
- **Operations time:** the time between the first and the last failure where the equipment was not in the out of production state.
- **SDT preventive maintenance:** the total time the equipment spent in preventive maintenance states. In our case the total time of scheduled downtime.
- **# of PM events:** the number of the preventive maintenance events.
- **UDT:** total time the equipment spent in the unscheduled downtime state.
- **# of failures:** the number of failures. Is equal to the number of unscheduled life cycles.

- **UDT repair, equipment-related:** total time the equipment spent in the [redacted] state.
- **# of equipment-related failures:** the number of equipment-related failures. Is equal to the number of equipment-related unscheduled life cycles.
- **Downtime:** total time the equipment spent in the downtime state.
- **# of continuous downtime events:** number of times the equipment went from uptime to downtime.

The codes can be found in appendix B.

3.4 Equipment

The equipment we chose to work on came from the wafer fab. For most of the cases we chose a couple of the similar equipment from the most value adding procedures in the fab that also happen to be the bottlenecks depending on the product mix. Also, all the equipment come from a clean room environment that increases the importance for high uptime as the space for the equipment is limited and the repairs are harder to complete. However, the analysis can apply to any equipment from the plant. Table 2 lists the equipment that was analysed, with some comments wherever appropriate.

Table 2 Equipment analysed

[redacted]	ProdArea	Eqptype	Type	# of cycles	Comments
[redacted]	[redacted]	[redacted]	ainframe	[redacted]	[redacted] is in a series with [redacted]. Lots start at [redacted] follow through to [redacted] to return again to [redacted] to later visit other equipment. [redacted] and [redacted] are photolithography equipment, a very important step in the process chain. The production area is called [redacted] and it is the new wafer fab of the plant.
[redacted]	[redacted]	[redacted]	ainframe	[redacted]	See [redacted].
[redacted]	[redacted]	[redacted]	ainframe	[redacted]	The same equipment as [redacted]. It is paired with [redacted].
[redacted]	[redacted]	[redacted]	ainframe	[redacted]	See [redacted].
[redacted]	[redacted]	[redacted]	ainframe	[redacted]	[redacted] is also a photolithography equipment but it is located in production area [redacted] (old wafer fab)
[redacted]	[redacted]	[redacted]	ainframe	[redacted]	The same equipment as [redacted].
[redacted]	[redacted]	[redacted]	ainframe	[redacted]	Chemical vapor deposition equipment. It is one of the most value-adding processes.
[redacted]	[redacted]	[redacted]	ainframe	[redacted]	Same as [redacted].
[redacted]	[redacted]	[redacted]	ainframe	[redacted]	Diffusion equipment. Also, one of the most value-adding processes.
[redacted]	[redacted]	[redacted]	Chamber	[redacted]	One of the chambers of [redacted]. If the chamber is down, the mainframe can still work and support the rest of the chambers. If the mainframe is down, the chamber is forced to be blocked.
[redacted]	[redacted]	[redacted]	ainframe	[redacted]	
[redacted]	[redacted]	[redacted]	ainframe	[redacted]	
[redacted]	[redacted]	[redacted]	ainframe	[redacted]	
[redacted]	[redacted]	[redacted]	ainframe	[redacted]	
[redacted]	[redacted]	[redacted]	ainframe	[redacted]	Equipment with many failures.
[redacted]	[redacted]	[redacted]	ainframe	[redacted]	Equipment with many failures.

4 E10 Metrics

The SEMI E10 standards provide a set of useful metrics to easily compere equipment and come to useful conclusions. We calculate the following metrics.

4.1 Metrics

1) $MTBF$— Mean time between failures; the average time between failures; total time divided by the number of failures during that time.

$$MTB\mathrm{F} = \frac{Total\ time}{\#\ of\ failures}$$

2) $E - MTBF$— Mean time between equipment-related failures; the average time between equipment-related failures; total time divided by the number of equipment-related failures during that time.

$$E - MTB\mathrm{F} = \frac{Total\ time}{\#\ of\ \text{equipment} - \text{related}\ failures}$$

3) $MTBF_U$ — Mean (uptime) time between failures; the average time equipment uptime between failures; uptime divided by the number of failures during that time. Only uptime is included in this calculation. Failures that occur coincident with an attempt made to change from non-scheduled time or scheduled downtime to an uptime state are included in this calculation.

$$MTBF_U = \frac{Uptime}{\#\ of\ failures\ during\ uptime}$$

4) MFD_U — Mean failure duration for failures during uptime; unscheduled downtime for failures during uptime divided by the number of failures during uptime.

$$MFD_U = \frac{Unscheduled\ downtime\ for\ failures\ during\ uptime}{\#\ of\ failures\ during\ uptime}$$

5) $E - MTBF_U$ — Mean (uptime) time between equipment-related failures; the average time the equipment performed its intended function between these equipment-related failures; uptime divided by the number of equipment-related failures during that time. Only uptime is included in this calculation. Equipment-related failures that occur coincident with an attempt made to change from non-scheduled time or scheduled downtime to an uptime state are included in this calculation.

$$E - MTBF_U = \frac{Uptime}{\#\ of\ equipment - related\ failures\ during\ uptime}$$

6) $MTBF_P$ — Mean (productive) time between failures; the average time the equipment performed its intended function between failures; productive time divided by the number of failures during that time. Only productive time is included in this calculation. Failures that occur when an attempt is made to change from any state other than unscheduled downtime to a productive state are included in this calculation.

$$MTBF_P = \frac{Productive\ time}{\#\ failures\ during\ productive\ time}$$

7) MFD_P — Mean failure duration for failures during productive time; unscheduled downtime for failures during productive time divided by the number of failures during productive time.

$$MFD_P = \frac{Unscheduled\ downtime\ during\ productive\ time}{\#\ failures\ during\ productive\ time}$$

8) $E - MTBF_P$ — Mean (productive) time between equipment-related failures; the average time the equipment performed its intended function between equipment-related failures; productive time divided by the number of equipment-related failures during that time. Only productive time is included in this calculation. Equipment-related failures that occur when an attempt is made to change from any state other than unscheduled time to a productive state are included in this calculation. Using $E - MTBF_P$, therefore, requires that the user not only have the capability of capturing failure information, but also tracking and categorizing total time and the root causes of failures accurately.

$$E - MTBF_P = \frac{Productive\ time}{\#\ \ equipment - related\ failures\ during\ productive\ time}$$

9) $Total\ uptime$ — The percent of time the equipment is in a condition to perform its intended function during total time.

$$Total\ uptime(100\%) = \frac{Uptime \times 100}{Total\ time}$$

10) $Operational\ uptime$ — The percent of time the equipment is in a condition to perform its intended function during the period of operations time and disregarding non-scheduled time.

$$Operational\ uptime(\%) = \frac{Uptime \times 100}{Operations\ time}$$

11) $MTTPM$ — Mean time to PM; the average time to complete a PM and return the equipment to a condition where it can perform its intended function; the sum of all PM time (elapsed time, not necessarily total man hours) incurred during a specified time period (including equipment and process test time, but not including maintenance delay downtime), divided by the number of PMs during that period.

$$MTTPM = \frac{SDT\ preventive\ maintenance}{\#\ of\ PM\ events}$$

12) $MTTR$ — Mean time to repair; the average time to correct a failure and return the equipment to a condition where it can perform its intended function; the sum of all repair time (elapsed time not necessarily total man hours) incurred during a specified time period (including equipment and process test time, but not including maintenance delay downtime), divided by the number of failures during that period.

$$MTTR = \frac{UDT}{\#\ of\ failures}$$

13) $E - MTTR$ — Mean time to repair equipment-related failures; the average time to correct an equipment-related failure and return the equipment to a condition where it can perform its intended function; the sum of all equipment-related failure repair time (elapsed time, not necessarily total man hours) incurred during a specified time period (including equipment and process test time, but not including maintenance delay downtime), divided by the number of equipment-related failures during that period.

$$E - MTTR = \frac{UDT\ repair, equipment - related}{\#\ of\ equipment - related\ failures}$$

14) $MTOL$ —Mean Time Off-Line; the average time equipment is not available to perform its intended function due to scheduled and unscheduled downtime combined. MTOL is the sum of

all downtime (including maintenance delay time) divided by the number of continuous downtime events during an observation period.

$$MTOL = \frac{Downtime}{\#\, of\ continuous\ downtime\ events}$$

15) $Total\, utilization$ — The percent of productive time during total time. This calculation is intended to reflect bottom-line equipment utilization.

$$Total\, utilization(\%) = \frac{Productive\ time \times 100}{Total\ time}$$

16) $Operational\, utilization$ — The percent of productive time during operations time. This calculation is intended to be used for equipment utilization comparisons between operations with different work shift configurations, since it does not include non-scheduled time.

$$Operational\, utilization(\%) = \frac{Productive\ time \times 100}{Operations\ time}$$

[6]

4.2 Results

Table 3 SEMI E10 metrics I.

[redacted]	MTBF	E-MTBF	MTBFu	MFDu	E-MTBFu	MTBFp	MFDp	E-MTBFp
[redacted]	[redacted]							
[redacted]								
[redacted]								
[redacted]								
[redacted]								
[redacted]								
[redacted]								
[redacted]								
[redacted]								
[redacted]								
[redacted]								
[redacted]								
[redacted]								
[redacted]								
[redacted]								
[redacted]								

Table 4 SEMI E10 metrics II.

[redacted]	Total Uptime (%)	Operational Uptime (%)	MTTR	MTTPM	E-MTTR	MTOL	Total Utilization (%)	Operational Utilization (%)
[redacted]	[redacted]							
[redacted]								
[redacted]								
[redacted]								
[redacted]								
[redacted]								
[redacted]								
[redacted]								
[redacted]								
[redacted]								
[redacted]								
[redacted]								
[redacted]								
[redacted]								
[redacted]								
[redacted]								

4.3 Conclusions

From the metrics we can come to several conclusions. Some are simple like the plant has shifts 24 hours a day,7 days a week, we can see that from the equal values the total and operations uptime have. In average the uptime percentages look good except for the few equipment in the end of the matrix. The problem should be the important MTTR values in combination with the frequent failures (small MTBF).

Another equipment with a problem are [redacted] and [redacted] with operational utilization close to zero. The reason for such utilization is the very small process time ([redacted] and [redacted] seconds respectively computed from the [redacted] table, maybe a mistake in the input system) in comparison with their pair equipment [redacted] and [redacted] that have long process time ([redacted] and [redacted] seconds respectively). So, the [redacted] and [redacted] control the rate the equipment pairs processes lots.

By compering MTBF with E-MTBF we see that for many equipment the values are close which means that the equipment does not stop producing from easily prevented failures like support material needed. [redacted] and [redacted] have a significant difference, it may be the masks needed to design the new wafers' layer.

We also see that [redacted] and [redacted] have large MFD_u values and checking MTTR and MTTPM we conclude that it is hard to repair the specific equipment in whole. It is very important to prevent a failure and we can see that both have very few failures ([redacted] and [redacted] respectively).

And finally, we see that $MTBF_u * Operations\ Utilization \approx MTBF_p$ that tells us our calculation are correct. There is more we can conclude from the results and if we had calculated more equipment we could come to many more conclusions with better accuracy.

5 Statistical distributions

5.1 Process steps

With the periods of time that we calculated as described in the previous chapter we search if and which statistical distribution they follow. We work in the following way:

1. Create histograms for the uptimes, processing times, productive times, downtimes, and times to restore.
2. Fit different probability distributions to the data and select the best one.
3. Perform a Kolmogorov-Smirnov goodness-of-fit statistical test for the best fitted distribution.

The bin size for the creation of the histograms was computed using Sturge's formula,

$K = 1 + 3.322\log_{N}$, where

$K =$ Number of bins (class intervals),

$N =$ Number of observations in the set.

The candidate statistical distributions considered are:

- Beta,
- Birnbaum-Saunders,
- Exponential,
- Extreme value,
- Gamma,
- Generalized extreme value,
- Generalized Pareto,
- Inverse Gaussian,
- Logistic,
- Log-logistic,
- Lognormal,
- Nakagami,
- Normal,
- Rayleigh,
- Rician,
- t-location scale,
- Weibull.

A program written in MATLAB was used to fit the above distributions to the data, based on maximum likelihood estimation. For each distribution, the program yielded the distribution parameters resulting in the best fit and the resulting errors.

To select the best distribution model, the program uses the Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC) that estimate the quality of each model, relative to each of the other models. Given a set of candidate models for the data, the preferred model is the one with the minimum AIC (or BIC) value. The AIC value of a model is given by $AIC = 2k - 2ln(\hat{L})$, where k is the number of estimated parameters and $L\hat{}$ is the maximum value of the likelihood function. Thus, AIC rewards goodness of fit (as assessed by the likelihood function), but it also includes a penalty that is an increasing function of the number of estimated parameters. The BIC value of a model is given by $BIC = ln(N)k - 2ln(L\hat{})$. The formula for BIC is similar to the formula for AIC, but with a different penalty for the number of parameters.

AIC (and BIC) does not provide a test of a model in the sense of testing a null hypothesis. It tells nothing about the absolute quality of a model, only the quality relative to other models. Thus, if all the candidate models fit poorly, AIC will not give any warning of that. To find how well the best distribution fits the data, we performed a Kolmogorov-Smirnov goodness-of-fit statistical test. The Kolmogorov–Smirnov statistic quantifies the distance between the empirical distribution function of the sample and the cumulative distribution function of a reference distribution. This test produces a significance value (p-value) indicating how much the empirical distribution function of the sample diverges from the cumulative distribution function of the reference distribution. In order to accept the null hypothesis that the sample is drawn from the best distribution, the p-value must be larger than a given significance level α. In the tests that were performed, the significance level that were used is $\alpha = 0.05$.

We also plot the hazard function for each of the chosen distribution. The hazard function is defined by

$$h(t) = \frac{f(t)}{1 - F(t)}$$

The hazard function of a random time (e.g. uptime) indicates the rate of change of the probability this time will soon come to an end as a function of time. For example, the hazard rate of the uptime of an equipment indicates the rate of change of the probability that this equipment which will soon cease to be up, given that it has been up during the last t time units. A positive hazard rate means that as time passes it is more likely that the equipment which will soon cease to be up, whereas a negative hazard rate means that as time passes it is less likely that the equipment which will soon cease to be up.

The code can be found in the appendix B.

5.2 Useful statistical distributions

The statistical distributions we used in our reliability analysis are the following.

5.2.1 Weibull distribution

The Weibull distribution is one of the most widely used lifetime distributions in reliability engineering. It is a versatile distribution that can take on the characteristics of other types of distributions, based on the value of the shape parameter, σ.

The Weibull distribution is given by

$$f(x;\ \eta, \sigma) = \frac{\eta}{\sigma} \left(\frac{x}{\sigma}\right)^{\eta - 1} e^{-\left(\frac{x}{\sigma}\right)^{\eta}}$$

where the variable x and the parameters η and σ all are positive real numbers. The distribution is named after the Swedish physicist Waloddi Weibull (1887–1979) a professor at the Technical Highschool in Stockholm 1924–1953.

The parameter σ is simply a scale parameter and the variable y = x/σ has the distribution

$$g(y) = \eta\, y^{\eta-1}\, e^{-y^{\eta}}$$

In figure we show the distribution for a few values of η. For η < 1 the distribution has its mode at y = 0, at η = 1 it is identical to the exponential distribution, and for η > 1 the distribution has a mode at

$$x = \left(\frac{\eta-1}{\eta}\right)^{\frac{1}{\eta}}$$

which approaches x = 1 as η increases (at the same time the distribution gets more symmetric and narrower). [7]

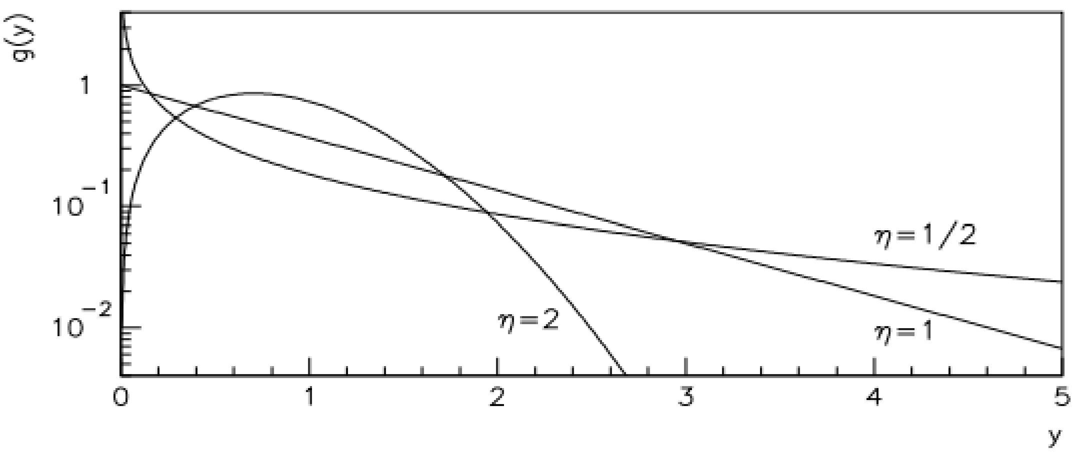


Figure 5.1 The Weibull distribution for different vales of the η parameter

5.2.2 Lognormal distribution

The lognormal distribution is used in reliability analysis of semiconductors and fatigue life of certain types of mechanical components. However, its main application is in maintainability analysis of time to repair data.

The log-normal distribution is given by

$$f(x;\ \mu, \sigma) = \frac{1}{x\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{\ln x-\mu}{\sigma}\right)^{2}}$$

where the variable $x > 0$ and the parameters μ and $\sigma > 0$ all are real numbers. It is sometimes denoted Λ(μ, σ^2) in the same spirit as we often denote a normally distributed variable by N(μ, σ^2).

If u is distributed as N(μ, σ^2) and $u = \ln x$ then x is distributed according to the *log-normal* distribution.

Note also that if x has the distribution Λ(μ, σ^2) then $y = e^{a}\, x^{b}$ is distributed as Λ($a + b$μ, $b^2\sigma^2$).

In figure we show the log-normal distribution for the basic form, with μ = 0 and σ = 1. [7]

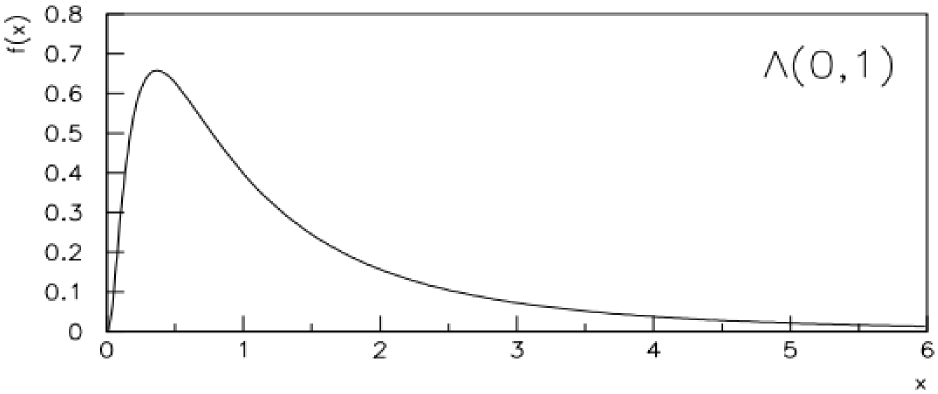


Figure 5.2 The Lognormal distribution

5.2.3 Exponential distribution

The exponential distribution is one of the widely used continuous distributions. It is often used to model the time elapsed between events.

The exponential distribution is given by

$$f(x;\ \alpha) = \frac{1}{\alpha}\ e^{-\frac{x}{\alpha}}$$

where the variable x as well as the parameter α is positive real quantities. The exponential distribution occurs in many different connections such as the radioactive or particle decays or the time between events in a Poisson process where events happen at a constant rate. [7]

5.2.4 Gamma distribution

The gamma distribution is a flexible life distribution model that may offer a good fit to some sets of failure data.

The Gamma distribution is given by

$$f(x;\ a,b)\ =\ a(ax)^{b-1}\ e^{-ax}/\Gamma(b)$$

where the parameters a and b are positive real quantities as is the variable x. Note that the parameter a is simply a scale factor.

For $b\ \leq\ 1$ the distribution is J-shaped and for b > 1 it is unimodal with its maximum at $x = \frac{b-1}{a}$.

In the special case where b is a positive integer this distribution is often referred to as the *Erlangian* distribution. For b = 1 we obtain the exponential distribution and with $a\ =\ 1/2$ and $b\ =\ n/2$ with n, an integer we obtain the chi-squared distribution with n degrees of freedom. In figure we show the Gamma distribution for b-values of 2 and 5. [7]

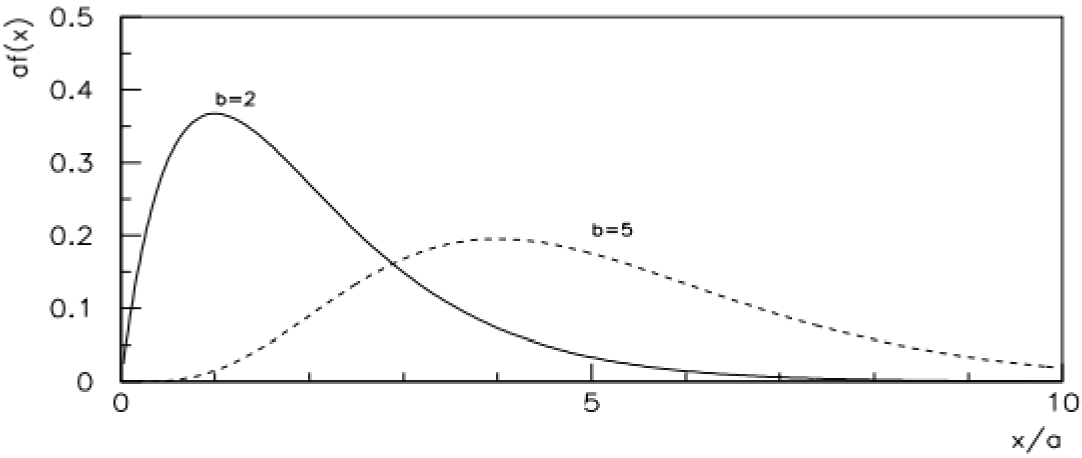


Figure 5.3 The Gamma distribution for different values of the b parameter.

5.2.5 Generalized Pareto distribution

In statistics, the generalized Pareto distribution (GPD) is a family of continuous probability distributions. It is often used to model the tails of another distribution. It is specified by three parameters: location μ, scale σ, and shape ξ. Sometimes it is specified by only scale and shape and sometimes only by its shape parameter. Some references give the shape parameter as $\kappa = -\xi$.

$$F(x;\ \mu,\sigma,\xi) = \frac{1}{\sigma}\left(1 + \xi\,\frac{x-\mu}{\sigma}\right)^{\frac{1}{\xi+1}}$$

5.3 Table with fitted distributions

In Table 20 we seen the results of the analysis. The columns are the following.

[redacted]**:** the equipment.

STATE: the state we fitted the distribution.

# of failures: the number of unscheduled downtime cycle.

Distribution: the chosen distribution.

Par1Dist: the description of the first parameter.

Par2Dist: the description of the second parameter.

Par3Dist: the description of the third parameter.

Par1: the first parameter.

Par2: the second parameter.

Par3: the third parameter.

Pvalue: the P value.

Hypot: 0 we accept the distribution and 1 we reject it.

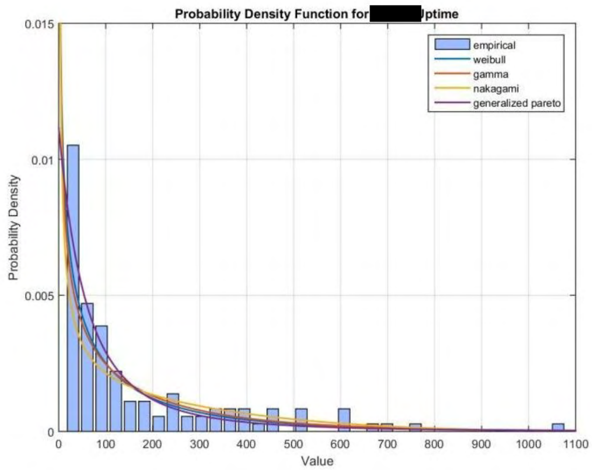
Table 5 Fitted distributions.

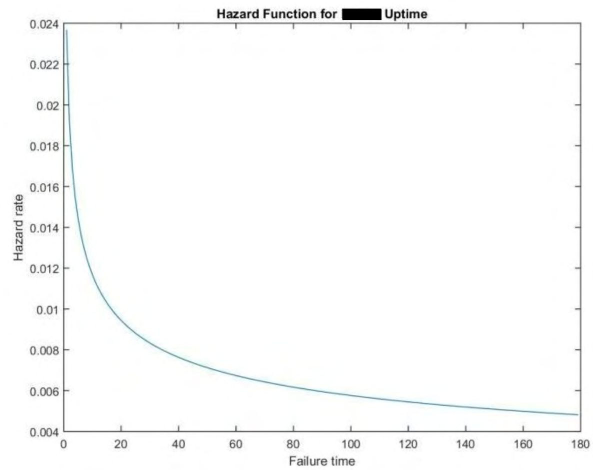
[redacted]	STATE	# of failures	Distribution	Par1Dist	Par2Dist	Par3Dist	Par1	Par2	Par3	pvalue	hypot
	Uptime	[redacted]	gamma	shape	scale		[redacted]	[redacted]	[redacted]	0.10726	0
	Processing Time	[redacted]	generalized pareto	shape	scale	threshold	[redacted]	[redacted]	[redacted]	0.0020602	1
[redacted]	Productive Time	[redacted]	generalized pareto	shape	scale	threshold	[redacted]	[redacted]	[redacted]	0.0015228	1
	Downtime	[redacted]	generalized pareto	shape	scale	threshold	[redacted]	[redacted]	[redacted]	0.046679	1
	Time to Restore	[redacted]	generalized pareto	shape	scale	threshold	[redacted]	[redacted]	[redacted]	0.023358	1
	Uptime	[redacted]	gamma	shape	scale		[redacted]	[redacted]	[redacted]	0.24081	0
	Processing Time	[redacted]	generalized pareto	shape	scale	threshold	[redacted]	[redacted]	[redacted]	0.66777	0
[redacted]	Productive Time	[redacted]	generalized pareto	shape	scale	threshold	[redacted]	[redacted]	[redacted]	0.18657	0
	Downtime	[redacted]	lognormal	log location	log scale		[redacted]	[redacted]	[redacted]	0.85769	0
	Time to Restore	[redacted]	generalized pareto	shape	scale	threshold	[redacted]	[redacted]	[redacted]	0.92829	0
	Uptime	[redacted]	weibull	scale	shape		[redacted]	[redacted]	[redacted]	0.73123	0
	Processing Time	[redacted]	generalized pareto	shape	scale	threshold	[redacted]	[redacted]	[redacted]	0.20227	0
[redacted]	Productive Time	[redacted]	generalized pareto	shape	scale	threshold	[redacted]	[redacted]	[redacted]	0.32083	0
	Downtime	[redacted]	generalized pareto	shape	scale	threshold	[redacted]	[redacted]	[redacted]	0.4794	0
	Time to Restore	[redacted]	generalized pareto	shape	scale	threshold	[redacted]	[redacted]	[redacted]	0.011563	1
	Uptime	[redacted]	gamma	shape	scale		[redacted]	[redacted]	[redacted]	0.25105	0
	Processing Time	[redacted]	generalized pareto	shape	scale	threshold	[redacted]	[redacted]	[redacted]	0.20087	0
[redacted]	Productive Time	[redacted]	generalized pareto	shape	scale	threshold	[redacted]	[redacted]	[redacted]	0.53796	0
	Downtime	[redacted]	lognormal	log location	log scale		[redacted]	[redacted]	[redacted]	0.30993	0
	Time to Restore	[redacted]	lognormal	log location	log scale		[redacted]	[redacted]	[redacted]	0.8027	0
	Uptime	[redacted]	weibull	scale	shape		[redacted]	[redacted]	[redacted]	0.92872	0
	Processing Time	[redacted]	weibull	scale	shape		[redacted]	[redacted]	[redacted]	0.049492	1
[redacted]	Productive Time	[redacted]	generalized pareto	shape	scale	threshold	[redacted]	[redacted]	[redacted]	0.12424	0
	Downtime	[redacted]	lognormal	log location	log scale		[redacted]	[redacted]	[redacted]	0.079721	0
	Time to Restore	[redacted]	lognormal	log location	log scale		[redacted]	[redacted]	[redacted]	0.66877	0
	Uptime	[redacted]	weibull	scale	shape		[redacted]	[redacted]	[redacted]	0.4794	0
	Processing Time	[redacted]	generalized pareto	shape	scale	threshold	[redacted]	[redacted]	[redacted]	0.018669	1
[redacted]	Productive Time	[redacted]	generalized pareto	shape	scale	threshold	[redacted]	[redacted]	[redacted]	0.10042	0
	Downtime	[redacted]	lognormal	log location	log scale		[redacted]	[redacted]	[redacted]	0.4794	0
	Time to Restore	[redacted]	lognormal	log location	log scale		[redacted]	[redacted]	[redacted]	0.84871	0
	Uptime	[redacted]	weibull	scale	shape		[redacted]	[redacted]	[redacted]	0.16531	0
	Processing Time	[redacted]	gamma	shape	scale		[redacted]	[redacted]	[redacted]	0.2864	0
[redacted]	Productive Time	[redacted]	gamma	shape	scale		[redacted]	[redacted]	[redacted]	0.87595	0
	Downtime	[redacted]	lognormal	log location	log scale		[redacted]	[redacted]	[redacted]	0.99992	0
	Time to Restore	[redacted]	generalized pareto	shape	scale	threshold	[redacted]	[redacted]	[redacted]	0.78159	0
	Uptime	[redacted]	weibull	scale	shape		[redacted]	[redacted]	[redacted]	0.7727	0
	Processing Time	[redacted]	generalized pareto	shape	scale	threshold	[redacted]	[redacted]	[redacted]	0.16436	0
[redacted]	Productive Time	[redacted]	generalized pareto	shape	scale	threshold	[redacted]	[redacted]	[redacted]	0.27707	0
	Downtime	[redacted]	lognormal	log location	log scale		[redacted]	[redacted]	[redacted]	0.4995	0
	Time to Restore	[redacted]	lognormal	log location	log scale		[redacted]	[redacted]	[redacted]	0.4995	0
	Uptime	[redacted]	lognormal	log location	log scale		[redacted]	[redacted]	[redacted]	0.62161	0
	Processing Time	[redacted]	lognormal	log location	log scale		[redacted]	[redacted]	[redacted]	0.38728	0
[redacted]	Productive Time	[redacted]	lognormal	log location	log scale		[redacted]	[redacted]	[redacted]	0.98678	0
	Downtime	[redacted]	generalized pareto	shape	scale	threshold	[redacted]	[redacted]	[redacted]	0.38728	0
	Time to Restore	[redacted]	exponential	mean			[redacted]	[redacted]	[redacted]	0.86076	0
	Uptime	[redacted]	weibull	scale	shape		[redacted]	[redacted]	[redacted]	0.85802	0
	Processing Time	[redacted]	gamma	shape	scale		[redacted]	[redacted]	[redacted]	0.3831	0
[redacted]	Productive Time	[redacted]	gamma	shape	scale		[redacted]	[redacted]	[redacted]	0.85802	0
	Downtime	[redacted]	exponential	mean			[redacted]	[redacted]	[redacted]	0.70212	0
	Time to Restore	[redacted]	lognormal	log location	log scale		[redacted]	[redacted]	[redacted]	0.85802	0
	Uptime	[redacted]	weibull	scale	shape		[redacted]	[redacted]	[redacted]	0.27091	0
	Processing Time	[redacted]	generalized pareto	shape	scale	threshold	[redacted]	[redacted]	[redacted]	0.49209	0
[redacted]	Productive Time	[redacted]	generalized pareto	shape	scale	threshold	[redacted]	[redacted]	[redacted]	0.58178	0
	Downtime	[redacted]	generalized pareto	shape	scale	threshold	[redacted]	[redacted]	[redacted]	0.91762	0
	Time to Restore	[redacted]	generalized pareto	shape	scale	threshold	[redacted]	[redacted]	[redacted]	0.91762	0
	Uptime	[redacted]	weibull	scale	shape		[redacted]	[redacted]	[redacted]	0.51046	0
	Processing Time	[redacted]	generalized pareto	shape	scale	threshold	[redacted]	[redacted]	[redacted]	0.043055	1
[redacted]	Productive Time	[redacted]	generalized pareto	shape	scale	threshold	[redacted]	[redacted]	[redacted]	0.19597	0
	Downtime	[redacted]	weibull	scale	shape		[redacted]	[redacted]	[redacted]	0.76519	0
	Time to Restore	[redacted]	generalized pareto	shape	scale	threshold	[redacted]	[redacted]	[redacted]	0.95495	0
	Uptime	[redacted]	weibull	scale	shape		[redacted]	[redacted]	[redacted]	0.93035	0
	Processing Time	[redacted]	exponential	mean			[redacted]	[redacted]	[redacted]	0.12162	0
[redacted]	Productive Time	[redacted]	exponential	mean			[redacted]	[redacted]	[redacted]	0.34412	0
	Downtime	[redacted]	weibull	scale	shape		[redacted]	[redacted]	[redacted]	0.6449	0
	Time to Restore	[redacted]	lognormal	log location	log scale		[redacted]	[redacted]	[redacted]	0.93035	0
	Uptime	[redacted]	weibull	scale	shape		[redacted]	[redacted]	[redacted]	0.89101	0
	Processing Time	[redacted]	generalized pareto	shape	scale	threshold	[redacted]	[redacted]	[redacted]	0.29256	0
[redacted]	Productive Time	[redacted]	generalized pareto	shape	scale	threshold	[redacted]	[redacted]	[redacted]	0.47145	0
	Downtime	[redacted]	weibull	scale	shape		[redacted]	[redacted]	[redacted]	0.1677	0
	Time to Restore	[redacted]	lognormal	log location	log scale		[redacted]	[redacted]	[redacted]	0.99712	0
	Uptime	[redacted]	generalized pareto	shape	scale	threshold	[redacted]	[redacted]	[redacted]	0.045598	1
	Processing Time	[redacted]	generalized pareto	shape	scale	threshold	[redacted]	[redacted]	[redacted]	0.13017	0
[redacted]	Productive Time	[redacted]	generalized pareto	shape	scale	threshold	[redacted]	[redacted]	[redacted]	0.030948	1
	Downtime	[redacted]	generalized pareto	shape	scale	threshold	[redacted]	[redacted]	[redacted]	0.051668	0
	Time to Restore	[redacted]	generalized pareto	shape	scale	threshold	[redacted]	[redacted]	[redacted]	0.0001529	1
	Uptime	[redacted]	lognormal	log location	log scale		[redacted]	[redacted]	[redacted]	0.00033071	1
	Processing Time	[redacted]	generalized pareto	shape	scale	threshold	[redacted]	[redacted]	[redacted]	3,54E-02	1
[redacted]	Productive Time	[redacted]	generalized pareto	shape	scale	threshold	[redacted]	[redacted]	[redacted]	4,49E-02	1
	Downtime	[redacted]	generalized pareto	shape	scale	threshold	[redacted]	[redacted]	[redacted]	0.077401	0
	Time to Restore	[redacted]	lognormal	log location	log scale		[redacted]	[redacted]	[redacted]	0.13606	0

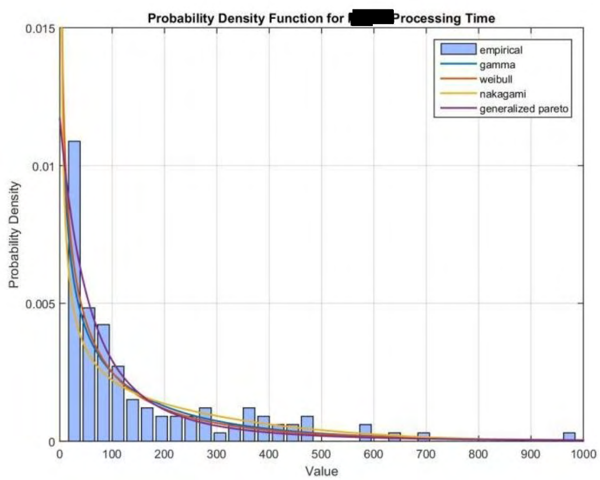
5.4 Histograms with fitted distributions

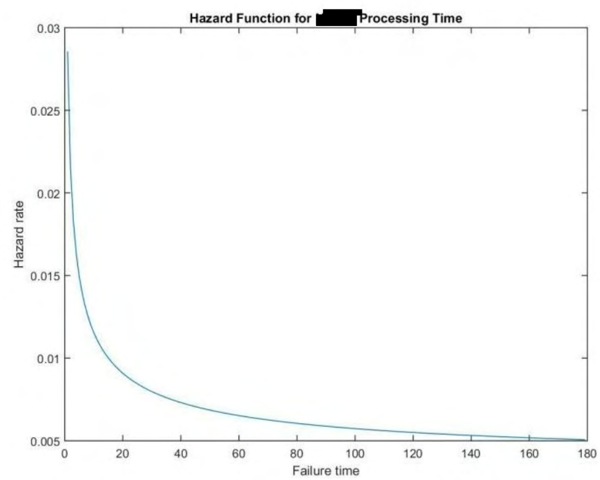
In this subsection you will be able to see the histograms from our samples with four fitted distribution where we chose one of them to test if it is accepted. Next to the histogram you will see the hazard function of the chosen fitted distribution. You will find only the equipment [redacted] here and the rest in the appendix A.

5.4.1 [redacted]









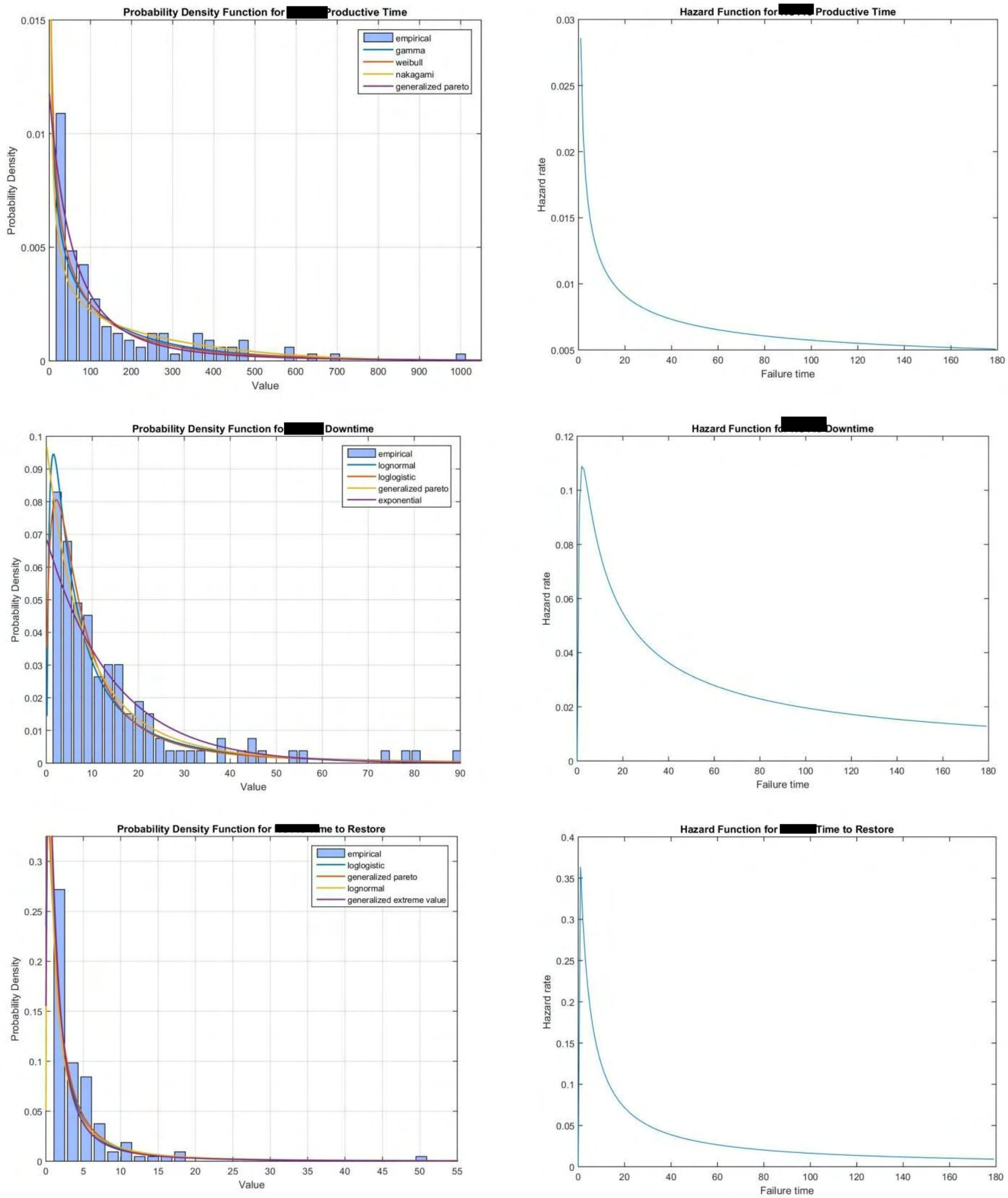


Figure 5.4 [redacted] Probability Density Function with Hazard Function for each set.

5.5 Conclusions

In results we see that we have a very good acceptance for our fitted distributions. In the eighty fitted distributions only fourteen failed to pass the P test with significance of a=0.05, if we chose to use a significance of a=0.01 we see that nine more would pass the test. The Downtime had the best acceptance with only one failed distribution, the Uptime had two failed distributions and three each for Production time and Time to Restore. The Processing time was the hardest to fit any distribution with five failed fitted distributions from sixteen equipment.

The Generalized Pareto is the most selected distribution with thirty-seven cases with Lognormal being the second most chosen with eighteen cases. Weibull distributions dominates in the uptime with ten cases but fails to live up to its expectations in reliability analyses with only four more cases in the rest. Gamma follows with seven cases and exponential impresses us with four cases against the other two and three parameter distributions.

An interesting result is that the hazard rates for the uptimes, processing times and productive times are decreasing. For the uptimes, (similarly for the processing times and productive times), this means that the more time has passed since the equipment was restored from an unscheduled downtime, the less likely it is that it will enter the unscheduled downtime state again. At first, this may seem surprising, because one would think that as time passes, the equipment becomes more fatigued and warn out from usage and therefore more likely to fail. What actually happens, however, is that as time passes, the equipment undergoes scheduled preventive maintenance and is therefore “rejuvenated”, meaning that it is less likely to fail after a maintenance event.

For the downtimes and the times to restore, the hazard rates are sharply increasing up to certain time and decreasing beyond that time. One possible explanation for this may be as follows. Under normal circumstances, the repair rates should be increasing, i.e. the longer a piece of equipment has been under repair, the higher the probability that it will soon be repaired. This is normal, because the repair process consists of a set of individual tasks that must be performed in a certain order, and so as the time to repair increases, more and more individual tasks are completed, and the entire repair process comes closer to completion. If, however, while the normal repair process is underway, it is discovered that the failure is more severe than what was originally thought, then more parts of the equipment need to be opened, more repair work need to be done, more specialized personnel need to be summoned, and new replacement parts need to be acquired. The same type of “over and above” findings can be repeated when the new parts of the equipment are opened, and so the probability that the repair will soon be completed may decrease as time passes.

The same analysis could happen also for the equipment-related unscheduled downtimes cycles.

6 Correlations

6.1 Process steps

In a complex environment like a factory many relationships between values are unknown, but that doesn't mean we can't suspect any. In this part of the analysis we check if there is any correlation between the following couples of values.

- **Processing time during a cycle vs. Maintenance time during the same cycle (PTvsMT)**
  One would expect that the more time is spent on maintenance of an equipment, the longer its processing time, due to the "rejuvenation" effect of the maintenance.
- **Processing time during a cycle vs. # of maintenance during the same cycle (PTvsNM)**
  One would expect that the higher the number of times an equipment is maintained, the longer its processing time, due to the "rejuvenation" effect of the maintenance.
- **Processing time during a cycle vs. Time to Restore from the next cycle (PTvsTTR)**
  It might not be unreasonable to expect that the longer the processing time of an equipment, the higher the accumulation of potential problems that could cause it to break down, and therefore the longer the time to restore the equipment once it finally breaks down.
- **Maintenance time during a cycle vs. Time to Restore from the next cycle (MTvsTTR)**
  It might not be unreasonable to expect that that the more time is spent on maintenance of an equipment before it breaks down, the more potential problems are caught before they cause a break down, and therefore the shorter the time to restore the equipment once it finally breaks down.
- **Autocorrelation of Time between Failures**
  Many analytical models of production systems rely on the assumption that the times between failures and the times to repair of the equipment are independent. Although lack of autocorrelation is necessary but not sufficient to show that successive observations of a random variable are independent, for the purposes of manufacturing management, testing for autocorrelation should suffice as an indication of independence.

We check for the correlations using MATLAB. We use Spearman correlation to find the relationship between the values and we also crate their scatter plot. We calculate also a P value with the hypothesis of being uncorrelated.

The code can be found in appendix B.

6.2 Results

In this table the columns are:

[redacted]: the machine.

VS: the correlated pair as described above.

Corr: the result of the spearman correlation.

Pval: the P value associated with the null hypothesis that the pairs are uncorrelated.

Hypo: is 0 if there we accept dependency and 1 if we reject dependency with a level of significance of $a = 0.05$.

Table 6 Correlations results

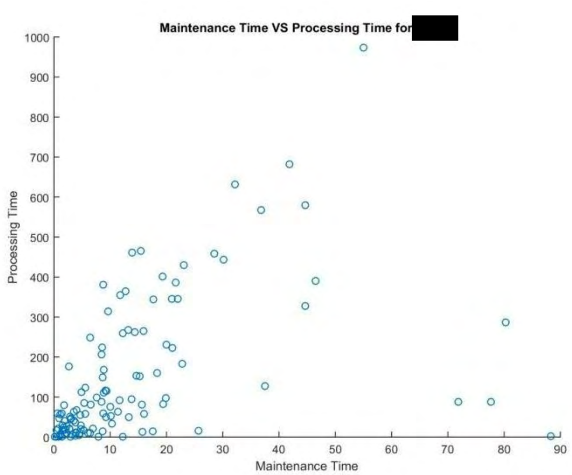
[redacted]	VS	CORR	PVAL	Hypo
[redacted]	PTvsMT	[redacted]	[redacted]	[redacted]
	PTvsNM	[redacted]	[redacted]	[redacted]
	PTvsTTR	[redacted]	[redacted]	[redacted]
	MTvsTTR	[redacted]	[redacted]	[redacted]
[redacted]	PTvsMT	[redacted]	[redacted]	[redacted]
	PTvsNM	[redacted]	[redacted]	[redacted]
	PTvsTTR	[redacted]	[redacted]	[redacted]
	MTvsTTR	[redacted]	[redacted]	[redacted]
[redacted]	PTvsMT	[redacted]	[redacted]	[redacted]
	PTvsNM	[redacted]	[redacted]	[redacted]
	PTvsTTR	[redacted]	[redacted]	[redacted]
	MTvsTTR	[redacted]	[redacted]	[redacted]
[redacted]	PTvsMT	[redacted]	[redacted]	[redacted]
	PTvsNM	[redacted]	[redacted]	[redacted]
	PTvsTTR	[redacted]	[redacted]	[redacted]
	MTvsTTR	[redacted]	[redacted]	[redacted]
[redacted]	PTvsMT	[redacted]	[redacted]	[redacted]
	PTvsNM	[redacted]	[redacted]	[redacted]
	PTvsTTR	[redacted]	[redacted]	[redacted]
	MTvsTTR	[redacted]	[redacted]	[redacted]
[redacted]	PTvsMT	[redacted]	[redacted]	[redacted]
	PTvsNM	[redacted]	[redacted]	[redacted]
	PTvsTTR	[redacted]	[redacted]	[redacted]
	MTvsTTR	[redacted]	[redacted]	[redacted]
[redacted]	PTvsMT	[redacted]	[redacted]	[redacted]
	PTvsNM	[redacted]	[redacted]	[redacted]
	PTvsTTR	[redacted]	[redacted]	[redacted]
	MTvsTTR	[redacted]	[redacted]	[redacted]
[redacted]	PTvsMT	[redacted]	[redacted]	[redacted]
	PTvsNM	[redacted]	[redacted]	[redacted]
	PTvsTTR	[redacted]	[redacted]	[redacted]
	MTvsTTR	[redacted]	[redacted]	[redacted]
[redacted]	PTvsMT	[redacted]	[redacted]	[redacted]
	PTvsNM	[redacted]	[redacted]	[redacted]
	PTvsTTR	[redacted]	[redacted]	[redacted]
	MTvsTTR	[redacted]	[redacted]	[redacted]
[redacted]	PTvsMT	[redacted]	[redacted]	[redacted]
	PTvsNM	[redacted]	[redacted]	[redacted]
	PTvsTTR	[redacted]	[redacted]	[redacted]
	MTvsTTR	[redacted]	[redacted]	[redacted]
[redacted]	PTvsMT	[redacted]	[redacted]	[redacted]
	PTvsNM	[redacted]	[redacted]	[redacted]
	PTvsTTR	[redacted]	[redacted]	[redacted]
	MTvsTTR	[redacted]	[redacted]	[redacted]
[redacted]	PTvsMT	[redacted]	[redacted]	[redacted]
	PTvsNM	[redacted]	[redacted]	[redacted]

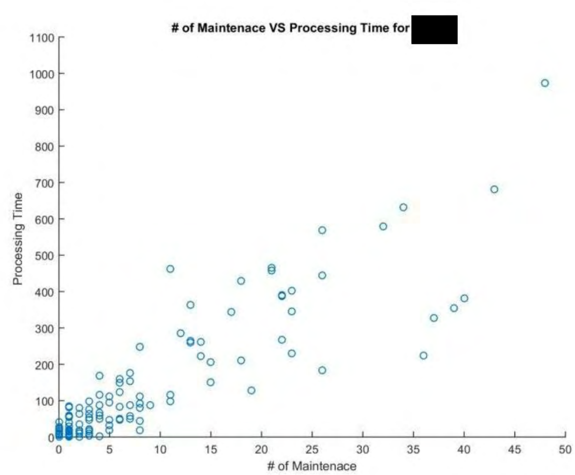
[redacted]	PTvsTTR	[redacted]
	MTvsTTR	[redacted]
[redacted]	PTvsMT	[redacted]
	PTvsNM	[redacted]
	PTvsTTR	[redacted]
	MTvsTTR	[redacted]
[redacted]	PTvsMT	[redacted]
	PTvsNM	[redacted]
	PTvsTTR	[redacted]
	MTvsTTR	[redacted]
[redacted]	PTvsMT	[redacted]
	PTvsNM	[redacted]
	PTvsTTR	[redacted]
	MTvsTTR	[redacted]
[redacted]	PTvsMT	[redacted]
	PTvsNM	[redacted]
	PTvsTTR	[redacted]
	MTvsTTR	[redacted]

6.3 Scatter plots

Here you will find the scatter plots of the correlation pairs and also the autocorrelations for the time between failures. For the autocorrelation the blue lines are the significant bounds to see if the autocorrelation is important, if the correlation is over the bound it is significant correlation for that specific lag. You will find only the equipment [redacted] here and the rest in the appendix A.

6.3.1 [redacted]





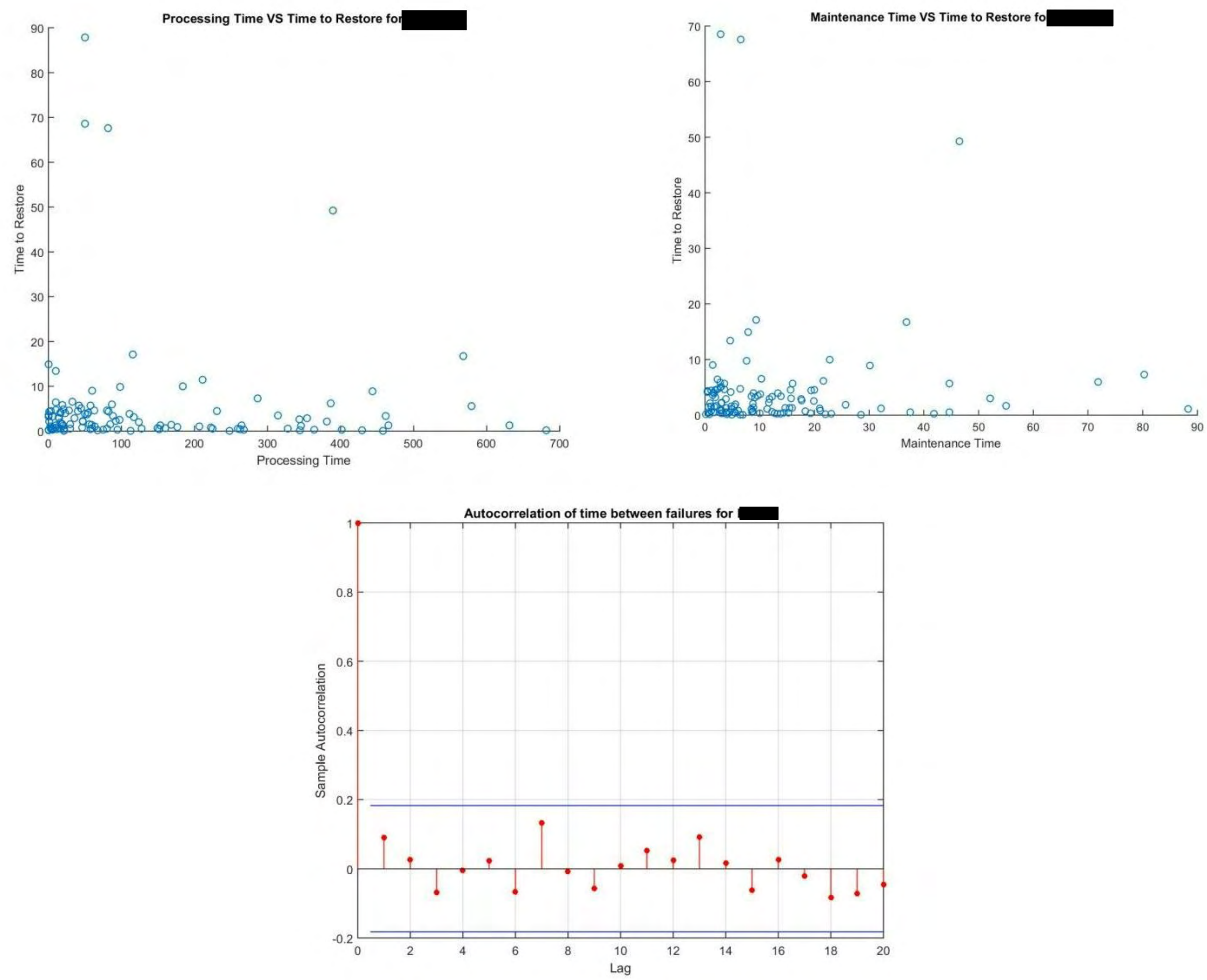


Figure 6.1 Correlations and autocorrelation for [redacted]

6.4 Conclusions

The results show us that there is a strong correlation of processing time with number of maintenances, that show us that every maintenance expends the processing time. Although it is possible to just show us that you can do more maintenances if the processing time is a longer period.

The rest of the correlations failed to show us any specific trend for the given pair of values. The processing time paired with maintenance time just gives us a hazy picture of the correlation between processing time with number of maintenances and for the other pairs any correlation seems like an outlier of the rule.

For the autocorrelation you do see some significant correlation in many cases but fail to see any repeating pattern. All of the significant autocorrelations are positive so longer time between failure mean longer time between failure after a specific amount of failures and shorter means shorter.

There are more pairs of correlations someone can check and depending on their experience can suspect better pairs.

7 Transition Rate matrix

7.1 Calculations

In many stochastic models of production systems, the operational state of each machine is modelled as a random process, most often a continuous-time Markov chain. Such a chain is fully characterized if we know the transition rates between the different states.

In this section, we present the transition rates that we calculated for each equipment that we studied. For simplification, we aggregated all the [redacted]-defined equipment states in Table 1, except NULL and [redacted], into 3 operations states, namely:

1 = Available ([redacted], [redacted]),

2 = Scheduled Downtime ([redacted], [redacted], [redacted]),

3 = Unscheduled Downtime ([redacted], [redacted], [redacted]),

The transition rate r_{ij} from each state i to each state j, $j \neq i$, $i,j = 1,2,3$, was calculated using the following expression:

$$r_{ij} = \frac{\text{number of transitions from } i \text{ to } j}{\text{total time}}$$

The self-transition rate r_{ii} for each state i, $i = 1,2,3$, can calculated using the following well-known expression for continuous-time Markov chains:

$$r_{ii} = -\sum_{j=1, j\neq i}^{3} r_{ij}$$

7.2 The transition rate matrix for each equipment

The following tables give the transition rates from each state i to each state j, $i,j = 1,2,3$, in a transition rate matrix form for each equipment. The units of these rates are in transitions per hour.

7.2.1 [redacted]

Table 7 Transition rate matrix for [redacted].

[redacted]

7.2.2 [redacted]

Table 8 Transition rate matrix for [redacted]

[redacted]

7.2.3 [redacted]

Table 9 Transition rate matrix for [redacted]

[redacted]

7.2.4 [redacted]

Table 10 Transition rate matrix for [redacted]

[redacted]

7.2.5 [redacted]

Table 11 Transition rate matrix for [redacted].

[redacted]

7.2.6 [redacted]

Table 12 Transition rate matrix for [redacted]

[redacted]

7.2.7 [redacted]

Table 13 Transition rate matrix for [redacted]

[redacted]

7.2.8 [redacted]

Table 14 Transition rate matrix for [redacted]

[redacted]

7.2.9 [redacted]

Table 15 Transition rate matrix for [redacted]

[redacted]

7.2.10 [redacted]

Table 16 Transition rate matrix for [redacted]

[redacted]

7.2.11 [redacted]

Table 17 Transition rate matrix for [redacted]

[redacted]

7.2.12 [redacted]

Table 18 Transition rate matrix for [redacted]

[redacted]

7.2.13 [redacted]

Table 19 Transition rate matrix for [redacted]

[redacted]

7.2.14 [redacted]

Table 20 Transition rate matrix for [redacted]

[redacted]

7.2.15 [redacted]

Table 21 Transition rate matrix for [redacted]

[redacted]

7.2.16 [redacted]

Table 22 Transition rate matrix for [redacted]

[redacted]

7.3 Conclusions

We see that the rates from 2 to 1 and 3 to 1 are bigger that 2 to 3 and 3 to 2 respectively which seems reasonable as you rarely see an equipment breaks down during a maintenance or to maintain one after you restore from a breakdown.

Also rates 1 to 2 are larger than 1 to 3 which tells us the transition from available to scheduled downtime is more often than the transition from available to unscheduled downtime and maybe also that the maintenance schedule prevents breakdowns.

8 Summary

Reviewing the whole course of the research we understand the value of the insights from the results. The SEMI E10 metrics gives us a good idea of the usage of the equipment, hints us of what to reconsider in the production and maintenance planning. The fitted distributions that have been calculated can be used in models that simulate the production line and with the correlation analysis we can also model the dependencies, used together we can form better strategies and production schedules. Also, the transition rate matrix can be used in continuous-time Markov chain models.

The results from the research give us many insights of the equipment and can also be used for further more analyses, however the most interesting part is the highly automated process that calculated all the results. It is possible to create a software to fit any production plant, not only coming from the semiconductor industry, with a few adjustments.

The work needed to complete this diploma thesis was a tough challenge however very rewarding. The real-life data revealed the size and difficulties the manufacturing plants have and the obstacles we need to surpass to come to any conclusion or suggestion.

Bibliography

[1] "A European co-funded innovation and lighthouse project on Digital Industry," [Online]. Available: https://productive40.eu/.

[2] "Semiconductor," [Online]. Available: https://en.wikipedia.org/wiki/Semiconductor.

[3] J. M. Fiore, "Semiconductor Devices Theory and Application," 3 April 2018. [Online]. Available: https://www2.mvcc.edu/users/faculty/jfiore/Linear/SemiconductorDevices.pdf. [Accessed 27 October 2018].

[4] L. Monch, J. W. Fowler and S. J. Mason, Production Planning and Control for Semiconductor Wafer Fabrication Facilities, vol. 52, S. Ramesh, Ed., Springer, 2013, pp. 11-13, 20-22.

[5] T. Pomorski, "Major Revision Update for SEMI E10 Specification for Definition and Measurement of Equipment Reliability, Availability, and Maintainability (RAM)," [Online]. Available: http://www.semi.org/en/semi-e10-specification-equipment-reliability-availability-and-maintainability. [Accessed 27 October 2018].

[6] M. Werre, W. Busing and I. McLeod, Background Statement for SEMI Draft Document 3846 info Revision to SEMI E10-0304E, SPECIFICATION FOR DEFINITION AND MEASUREMENT OF EQUIPMENT RELIABILITY, AVAILABILITY, AND MAINTAINABILITY (RAM) AND UTILIZATION, E. R. T. Force, Ed., San Jose: SEMI, 2011, pp. 2-13 , 18-24.

[7] C. Walck, "Hand-book on STATISTICAL DISTRIBUTIONS for experimentalists," University of Stockholm, Stockholm, 2007.

[8] [redacted], Anonymized [redacted] Data Description. [redacted]., 2018.

[9] V. H. Dhudshia, "SEMICON West 2006 STEP Methods to Measure/Improve Equipment Productivity," 12 July 2006. [Online]. Available: https://www.slideserve.com/samson/semicon-west-2006-step-methods-to-measure. [Accessed 27 October 2018].

[10] G. Liberopoulos and P. Tsarouhas, "Reliability Analysis of An Automated Pizza Production Line," *Journal of Food Engineering* , 2004.

[11] I. de Jong, R. Boumen, J. van de Mortel-Fronczak and J. Rooda, "Software reliability qualification for semi-conductor manufacturing systems," Eindhoven University of Technology research portal, Eindhoven, 2007.

[12] S. Munirathinam and B. Ramados, "Big Data Predictive Analtyics for Proactive Semiconductor Equipment Maintenance Review," Stanford University, California, 2014.

[13] D. J. Weidman, "Practical Reliability Engineering for Semiconductor Equipment," 14 January 2009. [Online]. Available: https://ewh.ieee.org/r1/boston/rl/files/boston_rs_meeting_jan09.pdf. [Accessed

27 October 2018].

[14] D. J. Weidman, “System Impact of Reliability,” MIT Lincoln Laboratory, Boston, 2015.

[15] Γ. Κωνσταντινίδης and Α. Μεσσής, “Ανάλυση Μεγάλων Δεδομένων Παραγωγής σε Βιομηχανία Κατασκευής Ημιαγωγών,” University of Thessaly, Volos, 2018.

Appendix B Codes

B.1 SQL Code

The code below selects the inputs we need from the databases. The equipment selected is the one stated on the second line and can easily be changed.

```
Declare @[redacted] nvarchar(30)
set [redacted]='[redacted]'
select * from [redacted]
where DATEDIFF(s,[redacted],[redacted])<>0 and [redacted]= [redacted]
order by [redacted];

select * from [redacted]
where [redacted]<>[redacted] AND [redacted]= [redacted]
order by [redacted];
```

B.2 Visual Basic Codes

B.2.1 Idle Time Spaces of Machine

The code below allocates the time spent in idle or productive, using the table [redacted] we extracted from the SQL database.

```
Sub idle()
   Dim lastrow As Long
   With Sheets("hist")
      lastrow = .Range("A" & .Rows.Count).End(xlUp).Row
   End With
   j = 1
   While j <= lastrow
      If Cells(j, 11) = "NULL" Then
          Rows(j).EntireRow.Delete
          j = j - 1
      End If
      j = j + 1
   Wend
   i = 1
   k = 1
   Cells(k, 16) = Cells(i, 10)
   Cells(k, 17) = Cells(i, 11)
   Cells(k, 18) = "[redacted]"
   With Sheets("hist")
      lastrow = .Range("A" & .Rows.Count).End(xlUp).Row
   End With
   For i = 2 To lastrow
      If Cells(i, 11) > Cells(k, 17) Then
         If Cells(i, 10) <= Cells(k, 17) Then
            Cells(k, 17) = Cells(i, 11)
         Else
```

```
            k = k + 1
            Cells(k, 16) = Cells(k - 1, 17)
            Cells(k, 17) = Cells(i, 10)
            Cells(k, 18) = "[redacted]"
            k = k + 1
            Cells(k, 16) = Cells(i, 10)
            Cells(k, 17) = Cells(i, 11)
            Cells(k, 18) = "[redacted]"
         End If
      End If
   Next i
End Sub
```

B.2.2 New Idle Equipment States Creation

The code below combines the results from the previous code with the HIST_EQUIPMENT_STATE_RtP table.

```
Sub tel()
Dim lastrow As Long
   With Sheets("eqps")
      lastrow = .Range("A" & .Rows.Count).End(xlUp).Row
   End With
'correct zero serial number
If Cells(1, 5) = 0 Then
   a = Cells(1, 1)
   b = Cells(1, 2)
   c = Cells(1, 3)
   d = Cells(1, 4)
   For i = 1 To lastrow - 1
      Cells(i, 1) = Cells(i + 1, 1)
      Cells(i, 2) = Cells(i + 1, 2)
      Cells(i, 3) = Cells(i + 1, 3)
      Cells(i, 4) = Cells(i + 1, 4)
      Cells(i, 5) = Cells(i + 1, 5)
   Next i
   Cells(lastrow, 1) = a
   Cells(lastrow, 2) = b
   Cells(lastrow, 3) = c
   Cells(lastrow, 4) = d
   Cells(lastrow, 5) = Cells(lastrow - 1, 5) + 1
End If
k = lastrow
i = 1
While i < lastrow
```

```
    j = 1
    e = 1
    If Cells(i, 4) = "[redacted]" Or Cells(i, 4) = "[redacted]" Then
        While Not (Cells(i, 2) >= Cells(j, 6) And Cells(i, 2) < Cells(j, 7))
            j = j + 1
        Wend
        While Not (Cells(i + 1, 2) >= Cells(e, 6) And Cells(i + 1, 2) < Cells(e, 7))
            e = e + 1
        Wend
        If j <> e Then
            Cells(i, 4) = Cells(j, 8)
            s = 0
            For t = j + 1 To e
                k = k + 1
                Cells(k, 1) = Cells(2, 1)
                Cells(k, 2) = Cells(t, 6)
                Cells(k, 3) = Cells(t - 1, 8)
                Cells(k, 4) = Cells(t, 8)
                s = s + 0.00001
                Cells(k, 5) = Cells(i, 5) + s
            Next t
            Cells(i + 1, 3) = Cells(e, 8)
        Else
             Cells(i, 4) = Cells(j, 8)
             Cells(i + 1, 3) = Cells(j, 8)
        End If
        i = i + 2
    Else
        i = i + 1
    End If
Wend
End Sub
```

B.2.3 E10 Times and Metrics Calculation

This is the last code before the statistical analysis begins. The code is calculating all the summary of time per cycle and all the metrics we are going to deal with, as they defined from E10 standard.

```
Sub data()
'deletezerotime
Dim lastrow As Long
With Sheets("eqps")
   lastrow = .Range("A" & .Rows.Count).End(xlUp).Row
End With
j = 1
While j <= lastrow - 1
   flag = False
   If Cells(j, 2) = Cells(j + 1, 2) Then
      Rows(j).EntireRow.Delete
      lastrow = lastrow - 1
      Cells(j, 3) = Cells(j - 1, 4)
      flag = True
   End If
   If flag = False Then
      j = j + 1
   End If
Wend
j = 1
While j <= lastrow
   flag = False
   If Cells(j, 4) = Cells(j, 3) Then
      Rows(j).EntireRow.Delete
      lastrow = lastrow - 1
      flag = True
   End If
   If flag = False Then
      j = j + 1
   End If
Wend
'test
j = 0
For i = 1 To lastrow - 1
   If Cells(i, 4) <> Cells(i + 1, 3) Then
      j = j + 1
   End If
Next
Cells(1, 1) = j
'availability
   i = 1
```

```
   While Cells(i, 4) <> "[redacted]" And Cells(i, 4) <> "[redacted]" And Cells(i, 4) <> "[redacted]" And i <= lastrow
      i = i + 1
   Wend
   i = i + 1
   Cells(1, 9) = "UPTIME"
   Cells(1, 16) = "NON-SCHEDULED TIME"
   k = 2
   If i <> lastrow Then
      While i < lastrow
         s = 0
         defunct = 0
         While Cells(i, 4) <> "[redacted]" And Cells(i, 4) <> "[redacted]" And Cells(i, 4) <> "[redacted]" And i < lastrow
            If Cells(i, 4) = "[redacted]" Or Cells(i, 4) = "[redacted]" Or Cells(i, 4) = "[redacted]" Then
               s = s + Cells(i + 1, 2) - Cells(i, 2)
            End If
            If Cells(i, 4) = "[redacted]" Then
               defunct = defunct + Cells(i + 1, 2) - Cells(i, 2)
            End If
            i = i + 1
         Wend
         If (Cells(i, 4) = "[redacted]" Or Cells(i, 4) = "[redacted]" Or Cells(i, 4) = "[redacted]") And Not (Cells(i - 1, 4) = "[redacted]" Or Cells(i - 1, 4) = "[redacted]" Or Cells(i - 1, 4) = "[redacted]") And i = lastrow Then
            Cells(k, 9) = s
            Cells(k, 16) = defunct
            k = k + 1
            If s = 0 Then
               Cells(k - 1, 13) = i
            End If
         End If
         If i < lastrow And Not (Cells(i - 1, 4) = "[redacted]" Or Cells(i - 1, 4) = "[redacted]" Or Cells(i - 1, 4) = "[redacted]") Then
            Cells(k, 9) = s
            Cells(k, 16) = defunct
            k = k + 1
            If s = 0 Then
               Cells(k - 1, 13) = i
            End If
         End If
         i = i + 1
      Wend
   End If
'reliability
   i = 1
```

```
  While Cells(i, 4) <> "[redacted]" And Cells(i, 4) <> "[redacted]" And Cells(i, 4) <> "[redacted]" And i <= lastrow
    i = i + 1
  Wend
  i = i + 1
  Cells(1, 10) = "PROCESSING TIME"
  k = 2
  If i <> lastrow Then
    While i < lastrow
      s = 0
      While Cells(i, 4) <> "[redacted]" And Cells(i, 4) <> "[redacted]" And Cells(i, 4) <> "[redacted]" And i < lastrow
        If Cells(i, 4) = "[redacted]" Or Cells(i, 4) = "[redacted]" Then
          s = s + Cells(i + 1, 2) - Cells(i, 2)
        End If
        i = i + 1
      Wend
      If (Cells(i, 4) = "[redacted]" Or Cells(i, 4) = "[redacted]" Or Cells(i, 4) = "[redacted]") And Not (Cells(i - 1, 4) = "[redacted]" Or Cells(i - 1, 4) = "[redacted]" Or Cells(i - 1, 4) = "[redacted]") And i = lastrow Then
        Cells(k, 10) = s
        k = k + 1
      End If
      If i < lastrow And Not (Cells(i - 1, 4) = "[redacted]" Or Cells(i - 1, 4) = "[redacted]" Or Cells(i - 1, 4) = "[redacted]") Then
        Cells(k, 10) = s
        k = k + 1
      End If
      i = i + 1
    Wend
  End If
'utilization
  i = 1
  While Cells(i, 4) <> "[redacted]" And Cells(i, 4) <> "[redacted]" And Cells(i, 4) <> [redacted]" And i <= lastrow
    i = i + 1
  Wend
  i = i + 1
  Cells(1, 11) = "PRODUCTIVE TIME"
  k = 2
  If i <> lastrow Then
    While i < lastrow
      s = 0
      While Cells(i, 4) <> "[redacted]" And Cells(i, 4) <> "[redacted]" And Cells(i, 4) <> "[redacted]" And i < lastrow
        If Cells(i, 4) = "[redacted]" Then
          s = s + Cells(i + 1, 2) - Cells(i, 2)
```

```
          End If
          i = i + 1
        Wend
        If (Cells(i, 4) = "[redacted]" Or Cells(i, 4) = "[redacted]" Or Cells(i, 4) = "[redacted]") And Not (Cells(i - 1, 4) = "[redacted]" Or Cells(i - 1, 4) = "[redacted]" Or Cells(i - 1, 4) = "[redacted]") And i = lastrow Then
          Cells(k, 11) = s
          k = k + 1
        End If
        If i < lastrow And Not (Cells(i - 1, 4) = "[redacted]" Or Cells(i - 1, 4) = "[redacted]" Or Cells(i - 1, 4) = "[redacted]") Then
          Cells(k, 11) = s
          k = k + 1
        End If
        i = i + 1
      Wend
   End If
'maintainability()
   i = 1
   While Cells(i, 3) = "[redacted]" And Cells(i, 3) = "[redacted]" And Cells(i, 3) = "[redacted]" And i < lastrow
      i = i + 1
   Wend
   While Cells(i, 4) <> "[redacted]" And Cells(i, 4) <> "[redacted]" And Cells(i, 4) <> "[redacted]" And i < lastrow
      i = i + 1
   Wend
   i = i + 1
   Cells(1, 12) = "DOWNTIME"
   k = 2
   If i <> lastrow Then
      s = 0
      While i < lastrow
        s = s + Cells(i, 2) - Cells(i - 1, 2)
        While Cells(i, 4) <> "[redacted]" And Cells(i, 4) <> "[redacted]" And Cells(i, 4) <> "[redacted]" And i < lastrow
          If Cells(i, 4) = "[redacted]" Or Cells(i, 4) = "[redacted]" Then
             s = s + Cells(i + 1, 2) - Cells(i, 2)
          End If
          i = i + 1
        Wend
        If (Cells(i, 4) = "[redacted]" Or Cells(i, 4) = "[redacted]" Or Cells(i, 4) = "[redacted]") And Not (Cells(i - 1, 4) = "[redacted]" Or Cells(i - 1, 4) = "[redacted]" Or Cells(i - 1, 4) = "[redacted]") And (i = lastrow) Then
          Cells(k, 12) = s
          k = k + 1
          s = 0
```

```
        End If
        If i < lastrow And Not (Cells(i - 1, 4) = "[redacted]" Or Cells(i - 1, 4) = "[redacted]" Or
Cells(i - 1, 4) = "[redacted]") Then
            Cells(k, 12) = s
            k = k + 1
            s = 0
        End If
        i = i + 1
      Wend
   End If
'only_scheduled_downtime
   i = 1
   While Cells(i, 4) <> "[redacted]" And Cells(i, 4) <> "[redacted]" And Cells(i, 4) <>
"[redacted]" And i <= lastrow
      i = i + 1
   Wend
   i = i + 1
   Cells(1, 13) = "MAITENANCE"
   k = 2
   If i <> lastrow Then
      While i < lastrow
        s = 0
        While Cells(i, 4) <> "[redacted]" And Cells(i, 4) <> "[redacted]" And Cells(i, 4) <>
"[redacted]" And i < lastrow
            If Cells(i, 4) = "[redacted]" Or Cells(i, 4) = "[redacted]" Then
               s = s + Cells(i + 1, 2) - Cells(i, 2)
            End If
            i = i + 1
        Wend
        If (Cells(i, 4) = "[redacted]" Or Cells(i, 4) = "[redacted]" Or Cells(i, 4) =
"[redacted]") And Not (Cells(i - 1, 4) = "[redacted]" Or Cells(i - 1, 4) = "[redacted]" Or
Cells(i - 1, 4) = "[redacted]") And (i = lastrow) Then
            Cells(k, 13) = s
            k = k + 1
        End If
        If i < lastrow And Not (Cells(i - 1, 4) = "[redacted]" Or Cells(i - 1, 4) = "[redacted]" Or
Cells(i - 1, 4) = "[redacted]") Then
            Cells(k, 13) = s
            k = k + 1
        End If
        i = i + 1
      Wend
   End If
'only_unscheduled_downtime
   i = 1
   While Cells(i, 3) = "[redacted]" And Cells(i, 3) = "[redacted]" And Cells(i, 3) =
"[redacted]" And i < lastrow
```

```
    i = i + 1
  Wend
  While Cells(i, 4) <> "[redacted]" And Cells(i, 4) <> "[redacted]" And Cells(i, 4) <>
"[redacted]" And i < lastrow
    i = i + 1
  Wend
  i = i + 1
  Cells(1, 14) = "TIME TO RESTORE"
  k = 2
  If i <> lastrow Then
    s = 0
    While i < lastrow
      s = s + Cells(i, 2) - Cells(i - 1, 2)
      While Cells(i, 4) <> "[redacted]" And Cells(i, 4) <> "[redacted]" And Cells(i, 4) <>
"[redacted]" And i < lastrow
        i = i + 1
      Wend
      If (Cells(i, 4) = "[redacted]" Or Cells(i, 4) = "[redacted]" Or Cells(i, 4) =
"[redacted]") And Not (Cells(i - 1, 4) = "[redacted]" Or Cells(i - 1, 4) = "[redacted]" Or
Cells(i - 1, 4) = "[redacted]") And (i = lastrow) Then
        Cells(k, 14) = s
        k = k + 1
        s = 0
      End If
      If i < lastrow And Not (Cells(i - 1, 4) = "[redacted]" Or Cells(i - 1, 4) = "[redacted]" Or
Cells(i - 1, 4) = "[redacted]") Then
        Cells(k, 14) = s
        k = k + 1
        s = 0
      End If
      i = i + 1
    Wend
  End If
'number_of_scheduled_downtime
  i = 1
  While Cells(i, 4) <> "[redacted]" And Cells(i, 4) <> "[redacted]" And Cells(i, 4) <>
"[redacted]" And i <= lastrow
    i = i + 1
  Wend
  i = i + 1
  Cells(1, 15) = "# OF MAINTENANCE"
  k = 2
  If i <> lastrow Then
    While i < lastrow
      s = 0
      While Cells(i, 4) <> "[redacted]" And Cells(i, 4) <> "[redacted]" And Cells(i, 4) <>
"[redacted]" And i < lastrow
```

```
          If Cells(i, 4) = "[redacted]" Or Cells(i, 4) = "[redacted]" Then
            While (Cells(i + 1, 4) = "[redacted]" Or Cells(i + 1, 4) = "[redacted]") And i < lastrow
              i = i + 1
            Wend
            s = s + 1
          End If
          i = i + 1
        Wend
        If (Cells(i, 4) = "[redacted]" Or Cells(i, 4) = "[redacted]" Or Cells(i, 4) =
"[redacted]") And Not (Cells(i - 1, 4) = "[redacted]" Or Cells(i - 1, 4) = "[redacted]" Or
Cells(i - 1, 4) = "[redacted]") And (i = lastrow) Then
          Cells(k, 15) = s
          k = k + 1
        End If
        If i < lastrow And Not (Cells(i - 1, 4) = "[redacted]" Or Cells(i - 1, 4) = "[redacted]" Or
Cells(i - 1, 4) = "[redacted]") Then
          Cells(k, 15) = s
          k = k + 1
        End If
        i = i + 1
      Wend
    End If
    'e-availability
    i = 1
    While Cells(i, 4) <> "[redacted]" And Cells(i, 4) <> "[redacted]" And Cells(i, 4) <>
"[redacted]" And i <= lastrow
      i = i + 1
    Wend
    i = i + 1
    Cells(1, 17) = "E-UPTIME"
    Cells(1, 24) = "E-NON-SCHEDULED TIME"
    k = 2
    If i <> lastrow Then
      While i < lastrow
        s = 0
        defunct = 0
        While Cells(i, 4) <> "[redacted]" And i < lastrow
          If Cells(i, 4) = "[redacted]" Or Cells(i, 4) = "[redacted]" Or Cells(i, 4) = "[redacted]" Then
            s = s + Cells(i + 1, 2) - Cells(i, 2)
          End If
          If Cells(i, 4) = "[redacted]" Then
            defunct = defunct + Cells(i + 1, 2) - Cells(i, 2)
          End If
          i = i + 1
        Wend
        If (Cells(i, 4) = "[redacted]") And i = lastrow Then
          Cells(k, 17) = s
```

```
            Cells(k, 24) = defunct
            k = k + 1
        End If
        If i < lastrow And Not (Cells(i - 1, 4) = "[redacted]") Then
            Cells(k, 17) = s
            Cells(k, 24) = defunct
            k = k + 1
        End If
        i = i + 1
    Wend
End If
'E-reliability
i = 1
While Cells(i, 4) <> "[redacted]" And i <= lastrow
    i = i + 1
Wend
i = i + 1
Cells(1, 18) = "E-PROCESSING TIME"
k = 2
If i <> lastrow Then
    While i < lastrow
        s = 0
        While Cells(i, 4) <> "[redacted]" And i < lastrow
            If Cells(i, 4) = "[redacted]" Or Cells(i, 4) = "[redacted]" Then
                s = s + Cells(i + 1, 2) - Cells(i, 2)
            End If
            i = i + 1
        Wend
        If (Cells(i, 4) = "[redacted]") And i = lastrow Then
            Cells(k, 18) = s
            k = k + 1
        End If
        If i < lastrow And Not (Cells(i - 1, 4) = "[redacted]") Then
            Cells(k, 18) = s
            k = k + 1
        End If
        i = i + 1
    Wend
End If
'E-utilization
i = 1
While Cells(i, 4) <> "[redacted]" And i <= lastrow
    i = i + 1
Wend
i = i + 1
Cells(1, 19) = "E-PRODUCTIVE TIME"
k = 2
```

```
    If i <> lastrow Then
        While i < lastrow
            s = 0
            While Cells(i, 4) <> "[redacted]" And i < lastrow
                If Cells(i, 4) = "[redacted]" Then
                    s = s + Cells(i + 1, 2) - Cells(i, 2)
                End If
                i = i + 1
            Wend
            If (Cells(i, 4) = "[redacted]") And i = lastrow Then
                Cells(k, 19) = s
                k = k + 1
            End If
            If i < lastrow And Not (Cells(i - 1, 4) = "[redacted]") Then
                Cells(k, 19) = s
                k = k + 1
            End If
            i = i + 1
        Wend
    End If
'E-maintainability()
    i = 1
    While Cells(i, 3) = "[redacted]" And i < lastrow
        i = i + 1
    Wend
    While Cells(i, 4) <> "[redacted]" And i < lastrow
        i = i + 1
    Wend
    i = i + 1
    Cells(1, 20) = "E-DOWNTIME"
    k = 2
    If i <> lastrow Then
        s = 0
        While i < lastrow
            s = s + Cells(i, 2) - Cells(i - 1, 2)
            While Cells(i, 4) <> "[redacted]" And i < lastrow
                If Cells(i, 4) = "[redacted]" Or Cells(i, 4) = "[redacted]" Or Cells(i, 4) =
[redacted]" Or Cells(i, 4) = "[redacted]" Then
                    s = s + Cells(i + 1, 2) - Cells(i, 2)
                End If
                i = i + 1
            Wend
            If (Cells(i, 4) = "[redacted]") And (i = lastrow) Then
                Cells(k, 20) = s
                k = k + 1
                s = 0
            End If
```

```
            If i < lastrow And Not (Cells(i - 1, 4) = "[redacted]") Then
                Cells(k, 20) = s
                k = k + 1
                s = 0
            End If
            i = i + 1
        Wend
    End If
'E-only_scheduled_downtime
    i = 1
    While Cells(i, 4) <> "[redacted]" And Cells(i, 4) <> "[redacted]" And Cells(i, 4) <>
"[redacted]" And i <= lastrow
        i = i + 1
    Wend
    i = i + 1
    Cells(1, 21) = "E-MAINTENANCE"
    k = 2
    If i <> lastrow Then
        While i < lastrow
            s = 0
            While Cells(i, 4) <> "[redacted]" And i < lastrow
                If Cells(i, 4) = "[redacted]" Or Cells(i, 4) = "[redacted]" Then
                    s = s + Cells(i + 1, 2) - Cells(i, 2)
                End If
                i = i + 1
            Wend
            If (Cells(i, 4) = "[redacted]") And (i = lastrow) Then
                Cells(k, 21) = s
                k = k + 1
            End If
            If i < lastrow And Not (Cells(i - 1, 4) = "[redacted]") Then
                Cells(k, 21) = s
                k = k + 1
            End If
            i = i + 1
        Wend
    End If
'E-only_unscheduled_downtime
    i = 1
    While Cells(i, 3) = "[redacted]" And i < lastrow
        i = i + 1
    Wend
    While Cells(i, 4) <> "[redacted]" And i < lastrow
        i = i + 1
    Wend
    i = i + 1
    Cells(1, 22) = "E-TIME TO RESTORE"
```

```
    k = 2
    If i <> lastrow Then
        s = 0
        While i < lastrow
            s = s + Cells(i, 2) - Cells(i - 1, 2)
            While Cells(i, 4) <> "[redacted]" And i < lastrow
                i = i + 1
            Wend
            If (Cells(i, 4) = "[redacted]") And (i = lastrow) Then
                Cells(k, 22) = s
                k = k + 1
                s = 0
            End If
            If i < lastrow And Not (Cells(i - 1, 4) = "[redacted]") Then
                Cells(k, 22) = s
                k = k + 1
                s = 0
            End If
            i = i + 1
        Wend
    End If
'E-number_of_scheduled_downtime
    i = 1
    While Cells(i, 4) <> "[redacted]" And i <= lastrow
        i = i + 1
    Wend
    i = i + 1
    Cells(1, 23) = "E-# OF MAINTENANCE"
    k = 2
    If i <> lastrow Then
        While i < lastrow
            s = 0
            While Cells(i, 4) <> "[redacted]" And i < lastrow
                If Cells(i, 4) = "[redacted]" Or Cells(i, 4) = "[redacted]" Or Cells(i, 4) =
"[redacted]" Or Cells(i, 4) = "[redacted]" Then
                    While (Cells(i + 1, 4) = "[redacted]" Or Cells(i + 1, 4) = "[redacted]" Or Cells(i, 4) =
"[redacted]" Or Cells(i, 4) = "[redacted]") And i < lastrow
                        i = i + 1
                    Wend
                    s = s + 1
                End If
                i = i + 1
            Wend
            If (Cells(i, 4) = "[redacted]") And Not (Cells(i - 1, 4) = "[redacted]" Or Cells(i - 1, 4) =
"[redacted]" Or Cells(i - 1, 4) = "[redacted]") And (i = lastrow) Then
                Cells(k, 23) = s
                k = k + 1
```

```
        End If
        If i < lastrow And Not (Cells(i - 1, 4) = "[redacted]") Then
            Cells(k, 23) = s
            k = k + 1
        End If
        i = i + 1
    Wend
End If
'validation
Cells(1, 26) = "UPTIME"
Cells(1, 27) = "PROCESSING TIME"
Cells(1, 28) = "PRODUCTIVE TIME"
Cells(1, 29) = "DOWNTIME"
Cells(1, 30) = "MAINTENANCE"
Cells(1, 31) = "TIME TO RESTORE"
Cells(1, 32) = "# OF MAINTENANCE"
Cells(1, 33) = "NON-SCHEDULED TIME"
Cells(1, 34) = "E-UPTIME"
Cells(1, 35) = "E-PROCESSING TIME"
Cells(1, 36) = "E-PRODUCTIVE TIME"
Cells(1, 37) = "E-DOWNTIME"
Cells(1, 38) = "E-MAINTENANCE"
Cells(1, 39) = "E-TIME TO RESTORE"
Cells(1, 40) = "E-# MAINTENANCE"
Cells(1, 41) = "E-NON-SCHEDULED TIME"
Dim lastrow2 As Long
    With Sheets("eqps")
        lastrow2 = .Range("I" & .Rows.Count).End(xlUp).Row
    End With
di = 0
For i = 2 To lastrow2
    If Cells(i, 9) < 0.005 Then
        Cells(i + 1, 9) = Cells(i + 1, 9) + Cells(i, 9)
        Cells(i + 1, 10) = Cells(i + 1, 10) + Cells(i, 10)
        Cells(i + 1, 11) = Cells(i + 1, 11) + Cells(i, 11)
        Cells(i + 1, 12) = Cells(i + 1, 12) + Cells(i, 12)
        Cells(i + 1, 13) = Cells(i + 1, 13) + Cells(i, 13)
        Cells(i + 1, 14) = Cells(i + 1, 14) + Cells(i, 14)
        di = di - 1
    Else
        Cells(i + di, 26) = Cells(i, 9) * 24
        Cells(i + di, 27) = Cells(i, 10) * 24
        Cells(i + di, 28) = Cells(i, 11) * 24
        Cells(i + di, 29) = Cells(i, 12) * 24
        Cells(i + di, 30) = Cells(i, 13) * 24
        Cells(i + di, 31) = Cells(i, 14) * 24
        Cells(i + di, 32) = Cells(i, 15)
```

```
        Cells(i + di, 33) = Cells(i, 16)
    End If
  Next
  Dim lastrow3 As Long
    With Sheets("eqps")
        lastrow3 = .Range("Q" & .Rows.Count).End(xlUp).Row
    End With
  di = 0
  For i = 2 To lastrow3
    If Cells(i, 17) < 0.005 Then
        Cells(i + 1, 17) = Cells(i + 1, 17) + Cells(i, 17)
        Cells(i + 1, 18) = Cells(i + 1, 18) + Cells(i, 18)
        Cells(i + 1, 19) = Cells(i + 1, 19) + Cells(i, 19)
        Cells(i + 1, 20) = Cells(i + 1, 20) + Cells(i, 20)
        Cells(i + 1, 21) = Cells(i + 1, 21) + Cells(i, 21)
        Cells(i + 1, 22) = Cells(i + 1, 22) + Cells(i, 22)

        di = di - 1
    Else
        Cells(i + di, 34) = Cells(i, 17) * 24
        Cells(i + di, 35) = Cells(i, 18) * 24
        Cells(i + di, 36) = Cells(i, 19) * 24
        Cells(i + di, 37) = Cells(i, 20) * 24
        Cells(i + di, 38) = Cells(i, 21) * 24
        Cells(i + di, 39) = Cells(i, 22) * 24
        Cells(i + di, 40) = Cells(i, 23)
        Cells(i + di, 41) = Cells(i, 24)
    End If
  Next
  i = 1
  While Cells(i, 3) = "[redacted]" And Cells(i, 3) = "[redacted]" And Cells(i, 3) =
"[redacted]" And i < lastrow
    i = i + 1
  Wend
  While Cells(i, 4) <> "[redacted]" And Cells(i, 4) <> "[redacted]" And Cells(i, 4) <>
"[redacted]" And i < lastrow
    i = i + 1
  Wend
  j = lastrow
  flag = False
  While j > 1 And flag = False
    While Cells(j, 4) <> "[redacted]" And Cells(j, 4) <> "[redacted]" And Cells(j, 4) <>
"[redacted]" And j > 1
        j = j - 1
    Wend
    If Cells(j, 3) <> "[redacted]" And Cells(j, 3) <> "[redacted]" And Cells(j, 3) <>
"[redacted]" Then
```

```
            flag = True
            j = j + 1
        End If
        j = j - 1
    Wend
    If i < j Then
        For k = 43 To 45
            Cells(1, k) = k - 42
            Cells(k - 41, 42) = k - 42
            Cells(2, k) = 0
            Cells(3, k) = 0
            Cells(4, k) = 0
        Next k
        For k = i + 1 To j
            If (Cells(k, 3) = "[redacted]" Or Cells(k, 3) = "[redacted]" Or Cells(k, 3) =
"[redacted]") And (Cells(k, 4) = "[redacted]" Or Cells(k, 4) = "[redacted]" Or Cells(k,
4) = "[redacted]") Then
                Cells(4, 44) = Cells(4, 44) + 1 / (24 * (Cells(j, 2) - Cells(i, 2)))
            ElseIf (Cells(k, 3) = "[redacted]" Or Cells(k, 3) = "[redacted]" Or Cells(k, 3) =
"[redacted]") And (Cells(k, 4) = "[redacted]" Or Cells(k, 4) = "[redacted]") Then
                Cells(4, 43) = Cells(4, 43) + 1 / (24 * (Cells(j, 2) - Cells(i, 2)))
            ElseIf (Cells(k, 3) = "[redacted]" Or Cells(k, 3) = "[redacted]" Or Cells(k, 3) = "[redacted]")
And (Cells(k, 4) = "[redacted]" Or Cells(k, 4) = "[redacted]") Then
                Cells(3, 43) = Cells(3, 43) + 1 / (24 * (Cells(j, 2) - Cells(i, 2)))
            ElseIf (Cells(k, 3) = "[redacted]" Or Cells(k, 3) = "[redacted]" Or Cells(k, 3) = "[redacted]")
And (Cells(k, 4) = "[redacted]" Or Cells(k, 4) = "[redacted]" Or Cells(k, 4) =
"[redacted]") Then
                Cells(3, 45) = Cells(3, 45) + 1 / (24 * (Cells(j, 2) - Cells(i, 2)))
            ElseIf (Cells(k, 3) = "[redacted]" Or Cells(k, 3) = "[redacted]") And (Cells(k, 4) = "[redacted]"
Or Cells(k, 4) = "[redacted]" Or Cells(k, 4) = "[redacted]") Then
                Cells(2, 45) = Cells(2, 45) + 1 / (24 * (Cells(j, 2) - Cells(i, 2)))
            ElseIf (Cells(k, 3) = "[redacted]" Or Cells(k, 3) = "[redacted]") And (Cells(k, 4) =
"[redacted]" Or Cells(k, 4) = "[redacted]" Or Cells(k, 4) = "[redacted]") Then
                Cells(2, 44) = Cells(2, 44) + 1 / (24 * (Cells(j, 2) - Cells(i, 2)))
            End If
        Next k
    End If
    Worksheets("eqps").Activate
With Sheets("eqps")
    lastrow = .Range("Z" & .Rows.Count).End(xlUp).Row
End With
'number of failures during productive time= nofdpt
For i = 2 To lastrow
    If Cells(i, 28) <> 0 Then
        nofdpt = nofdpt + 1
    End If
Next i
```

```
'uptime = ut
ut = 0
For i = 2 To lastrow
   ut = ut + Cells(i, 26)
Next i
'# of failures=nof
nof = lastrow - 1
'# of equipment related failures during uptime = noerfdu
With Sheets("eqps")
   lastrow2 = .Range("AH" & .Rows.Count).End(xlUp).Row
End With
noerfdu = lastrow2 - 1
'productive time = pt
pt = 0
For i = 2 To lastrow
   pt = pt + Cells(i, 28)
Next i
'# of equipment related failures during productive time = noerfdpt
noerfdpt = 0
For i = 2 To lastrow2
   If Cells(i, 36) <> 0 Then
      noerfdpt = noerfdpt + 1
   End If
Next i
'total time = tt
tt = 0
For i = 2 To lastrow
   tt = tt + Cells(i, 26) + Cells(i, 29) + Cells(i, 33)
Next i
'downtime = dt
For i = 2 To lastrow
   dt = dt + Cells(i, 29)
Next i
'operations time = ot
ot = ut + dt
'SDT preventive maintenance = sdtpm
sdtpm = 0
For i = 2 To lastrow
   sdtpm = sdtpm + Cells(i, 30)
Next i
'# of PM events = nopme
nopme = 0
For i = 2 To lastrow
   nopme = nopme + Cells(i, 32)
Next i
'unscheduled DT = udt
udt = 0
```

```
For i = 2 To lastrow
   udt = udt + Cells(i, 31)
Next i
'equipment related unscheduled DT = udt
erudt = 0
For i = 2 To lastrow2
   erudt = erudt + Cells(i, 39)
Next i
'# of continuous downtime events = nocde
nocde = 0
With Sheets("eqps")
   lastrow3 = .Range("A" & .Rows.Count).End(xlUp).Row
End With
i = 1
While (Cells(i, 3) = "[redacted]" Or Cells(i, 3) = "[redacted]" Or Cells(i, 3) =
"[redacted]") And i < lastrow3
   i = i + 1
Wend
While i < lastrow3
   If Cells(i, 4) = "[redacted]" Or Cells(i, 4) = "[redacted]" Or Cells(i, 4) = "[redacted]"
Or Cells(i, 4) = "[redacted]" Or Cells(i, 4) = "[redacted]" Then
      flag = False
      If Cells(i, 3) = "[redacted]" Or Cells(i, 3) = "[redacted]" Or Cells(i, 3) = "[redacted]" Then
         flag = True
      End If
      While (Cells(i, 4) = "[redacted]" Or Cells(i, 4) = "[redacted]" Or Cells(i, 4) =
"[redacted]" Or Cells(i, 4) = "[redacted]" Or Cells(i, 4) = "[redacted]") And i < lastrow3
         i = i + 1
      Wend
      If flag = True And i < lastrow3 Then
         nocde = nocde + 1
      End If
      If flag = True And i = lastrow3 And Cells(i, 4) <> "[redacted]" And Cells(i, 4) <>
"[redacted]" And Cells(i, 4) <> "[redacted]" And Cells(i, 4) <> "[redacted]"
And Cells(i, 4) <> "[redacted]" Then
         nocde = nocde + 1
      End If
   End If
   i = i + 1
Wend
Dim ws As Worksheet
   Set ws = ThisWorkbook.Sheets.Add(After:= _
         ThisWorkbook.Sheets(ThisWorkbook.Sheets.Count))
   ws.Name = "metrics"
Worksheets("metrics").Activate
Cells(1, 1) = "MTBFu"
Cells(2, 1) = ut / nof
```

```
Cells(1, 2) = "MFDu"
Cells(2, 2) = udt / nof
Cells(1, 3) = "E-MTBFu"
Cells(2, 3) = ut / noerfdu
Cells(1, 4) = "MTBFp"
Cells(2, 4) = pt / nofdpt
Cells(1, 5) = "E-MTBFp"
Cells(2, 5) = pt / noerfdpt
Cells(1, 6) = "Total Uptime"
Cells(2, 6) = ut / tt
Cells(1, 7) = "Operational Uptime"
Cells(2, 7) = ut / ot
Cells(1, 8) = "MTTR"
Cells(2, 8) = udt / nof
Cells(1, 9) = "MTTPM"
Cells(2, 9) = sdtpm / nopme
Cells(1, 10) = "E-MTTR"
Cells(2, 10) = erudt / noerfdu
Cells(1, 11) = "MTOL"
Cells(2, 11) = (dt + udt) / nocde
Cells(1, 12) = "Total Utilization"
Cells(2, 12) = pt / tt
Cells(1, 13) = "Operational Utilization"
Cells(2, 13) = pt / ot
Cells(1, 14) = "MFDp"
Cells(2, 14) = udt / nofdpt
Cells(1, 15) = "MTBF"
Cells(2, 15) = tt / nof
Cells(1, 16) = "E-MTBF"
Cells(2, 16) = tt / noerfdu
  Set ws = ThisWorkbook.Sheets.Add(After:= _
       ThisWorkbook.Sheets(ThisWorkbook.Sheets.Count))
  ws.Name = "data"
Worksheets("data").Activate
For i = 1 To 16
  For j = 1 To lastrow
    Cells(j, i) = Sheets("eqps").Cells(j, i + 25)
    Worksheets("data").Activate
  Next j
Next i
For i = 1 To 4
  For j = 1 To 4
    Cells(i, j + 16) = Sheets("eqps").Cells(i, j + 41)
    Worksheets("data").Activate
  Next j
Next i
Cells(1, 22) = "#failures"
```

```
Cells(2, 22) = nof
Cells(1, 23) = "#Efailures"
Cells(2, 23) = noerfdu
End Sub
```

B.3 MATLAB codes

B.3.1 Distributions

The following code finds a fitting distribution for the different values of the unscheduled downtime cycles and tests it if it a good match. It also plots the histograms with some good fitting distribution and the hazard function of the tested distribution.

```
i1=1;
i2=2;
time={'Uptime','Processing Time','Productive Time','Downtime','Time to
Restore'};
machines={'[redacted]','[redacted]','[redacted]','[redacted]','[redacted]','[redacted]','[redacted]', [redacted] ,'[redacted]','[redacted]','[redacted]','[redacted]','[redacted]','[redacted]','[redacted]','[redacted]'};
%
for i=machines

filename=strcat('C:\Users\spiros\Desktop\matlabthesis\equipment\',i(1),'.xlsm
');
    filename1=filename{1};
    nof=xlsread(filename1,'data','V2');
    maintenance=xlsread(filename1,'data',['G2:G',num2str(nof+1)]);
    metrics=xlsread(filename1,'metrics','A2:p2');

xlswrite('C:\Users\spiros\Desktop\matlabthesis\results.xlsx',[i,metrics(15),m
etrics(16),metrics(1),metrics(2),metrics(3),metrics(4),metrics(14),metrics(5)
,metrics(6),metrics(7),metrics(8),metrics(9),metrics(10),metrics(11),metrics(
12),metrics(13)],'metrics',['A',num2str(i2),':Q',num2str(i2)])
    i2=i2+1;
    i3=0;
    for j=[ 'A', 'B', 'C', 'D', 'F']
       i1=i1+1;
       i3=i3+1;


times=xlsread(filename1,'data',[num2str(j),'2:',num2str(j),num2str(nof+1)]);
       [P, D]=allfitdist(times);   %  ,'PDF'
       title(strjoin(['Probability Density Function for',i(1),time(i3)]))
       x1 = input('wait')

saveas(gcf,strjoin(['C:\Users\spiros\Desktop\matlabthesis\histographs\',i(1),
time(i3),'.jpg']))
       for k=1:4
       if  strcmp(P(k).DistName, 'weibull')
           R= wblrnd(P(k).Params(1),P(k).Params(2) ,[nof 1]);
           x = 0:1:metrics(15);
           Burrhazard = pdf('Weibull',x,P(k).Params(1),P(k).Params(2))./(1-
cdf('Weibull',x,P(k).Params(1),P(k).Params(2)));
           figure()
```

```
            plot(x,Burrhazard)
            title(strjoin(['Hazard Function for',i(1),time(i3)]));
            xlabel('Failure time');
            ylabel('Hazard rate');

saveas(gcf,strjoin(['C:\Users\spiros\Desktop\matlabthesis\histographs\',i(1),
time(i3),'hazard.jpg']))
            break
        elseif strcmp(P(k).DistName, 'lognormal')
            R = lognrnd(P(k).Params(1),P(k).Params(2) ,[nof 1]);
            x = 0:1:metrics(15);
            Burrhazard = pdf('Lognormal',x,P(k).Params(1),P(k).Params(2))./(1-
cdf('Lognormal',x,P(k).Params(1),P(k).Params(2)));
            figure()
            plot(x,Burrhazard)
            title(strjoin(['Hazard Function for',i(1),time(i3)]));
            xlabel('Failure time');
            ylabel('Hazard rate');

saveas(gcf,strjoin(['C:\Users\spiros\Desktop\matlabthesis\histographs\',i(1),
time(i3),'hazard.jpg']))
            break
        elseif strcmp(P(k).DistName, 'normal')
            R=normrnd(P(k).Params(1),P(k).Params(2) ,[nof 1]);
            x = 0:1:metrics(15);
            Burrhazard = pdf('Normal',x,P(k).Params(1),P(k).Params(2))./(1-
cdf('Normal',x,P(k).Params(1),P(k).Params(2)));
            figure()
            plot(x,Burrhazard)
            title(strjoin(['Hazard Function for',i(1),time(i3)]));
            xlabel('Failure time');
            ylabel('Hazard rate');

saveas(gcf,strjoin(['C:\Users\spiros\Desktop\matlabthesis\histographs\',i(1),
time(i3),'hazard.jpg']))
            break
        elseif strcmp(P(k).DistName, 'exponential')
            R= exprnd(P(k).Params(1) ,[nof 1]);
            x = 0:1:metrics(15);
            Burrhazard = pdf('Exponential',x,P(k).Params(1))./(1-
cdf('Exponential',x,P(k).Params(1)));
            figure()
            plot(x,Burrhazard)
            title(strjoin(['Hazard Function for',i(1),time(i3)]));
            xlabel('Failure time');
            ylabel('Hazard rate');

saveas(gcf,strjoin(['C:\Users\spiros\Desktop\matlabthesis\histographs\',i(1),
time(i3),'hazard.jpg']))
            break
        elseif strcmp(P(k).DistName, 'gamma')
            R= gamrnd(P(k).Params(1),P(k).Params(2) ,[nof 1]);
            x = 0:1:metrics(15);
            Burrhazard = pdf('Gamma',x,P(k).Params(1),P(k).Params(2))./(1-
cdf('Gamma',x,P(k).Params(1),P(k).Params(2)));
            figure()
            plot(x,Burrhazard)
```

```
            title(strjoin(['Hazard Function for',i(1),time(i3)]));
            xlabel('Failure time');
            ylabel('Hazard rate');

saveas(gcf,strjoin(['C:\Users\spiros\Desktop\matlabthesis\histographs\',i(1),
time(i3),'hazard.jpg']))
            break
        elseif strcmp(P(k).DistName, 'generalized pareto')
            R= gprnd(P(k).Params(1),P(k).Params(2) ,P(k).Params(3),[nof 1]);
            x = 0:1:metrics(15);
            Burrhazard = pdf('Generalized
Pareto',x,P(k).Params(1),P(k).Params(2),P(k).Params(3))./(1-cdf('Generalized
Pareto',x,P(k).Params(1),P(k).Params(2),P(k).Params(3)));
            figure()
            plot(x,Burrhazard)
            title(strjoin(['Hazard Function for',i(1),time(i3)]));
            xlabel('Failure time');
            ylabel('Hazard rate');

saveas(gcf,strjoin(['C:\Users\spiros\Desktop\matlabthesis\histographs\',i(1),
time(i3),'hazard.jpg']))
            break


        end
        end

        [h,p] = kstest2(times,R,'Alpha',0.05);

        if length(P(k).ParamDescription)==1

A={i{1},time{i3},num2str(nof),P(k).DistName,P(k).ParamDescription{1},' ' , '
' ,num2str(P(k).Params(1)),' ' ,' ', num2str(p),num2str(h)};
        elseif length(P(k).ParamDescription)==2

A={i{1},time{i3},num2str(nof),P(k).DistName,P(k).ParamDescription{1},P(k).Par
amDescription{2},' ' ,num2str(P(k).Params(1)),num2str(P(k).Params(2)), ' ',
num2str(p),num2str(h)};
        elseif length(P(k).ParamDescription)==3

A={i{1},time{i3},num2str(nof),P(k).DistName,P(k).ParamDescription{1},P(k).Par
amDescription{2},P(k).ParamDescription{3},
num2str(P(k).Params(1)),num2str(P(k).Params(2)),num2str(P(k).Params(3)),
num2str(p),num2str(h)};
        end

xlswrite('C:\Users\spiros\Desktop\matlabthesis\results.xlsx',A,'distributions
',['A',num2str(i1),':L',num2str(i1)])
        for k=1:10
          if  strcmp(P(k).DistName, 'weibull')
            R= wblrnd(P(k).Params(1),P(k).Params(2) ,[nof 1]);
            break
          elseif strcmp(P(k).DistName, 'exponential')
              P(k).Params(1)=P(k).Params(1);
              P(k).Params(2)=1;

              P(k).ParamDescription{2}=' ';
```

```
            R= wblrnd(P(k).Params(1),P(k).Params(2) ,[nof 1]);
            break
        end
      end
      [h,p] = kstest2(times,R,'Alpha',0.05);
      if P(k).Params(2)<1

A={i{1},time{i3},num2str(nof),P(k).DistName,P(k).ParamDescription{1},P(k).Par
amDescription{2} ,num2str(P(k).Params(1)),num2str(P(k).Params(2)),
num2str(p),num2str(h),'decreasing'};
      elseif P(k).Params(2)>1

A={i{1},time{i3},num2str(nof),P(k).DistName,P(k).ParamDescription{1},P(k).Par
amDescription{2} ,num2str(P(k).Params(1)),num2str(P(k).Params(2)),
num2str(p),num2str(h),' increasing'};
      else

A={i{1},time{i3},num2str(nof),P(k).DistName,P(k).ParamDescription{1},P(k).Par
amDescription{2} ,num2str(P(k).Params(1)),num2str(P(k).Params(2)),
num2str(p),num2str(h),'stable'};
      end

xlswrite('C:\Users\spiros\Desktop\matlabthesis\results.xlsx',A,'Weibull',['A'
,num2str(i1),':K',num2str(i1)])

    end
end
```

B.3.2 ALLFITDIST Function

The allfitdist function from Mike Sheppard, which we used in the above code.

```
function [D PD] = allfitdist(data,sortby,varargin)
%ALLFITDIST Fit all valid parametric probability distributions to data.
%   [D PD] = ALLFITDIST(DATA) fits all valid parametric probability
%   distributions to the data in vector DATA, and returns a struct D of
%   fitted distributions and parameters and a struct of objects PD
%   representing the fitted distributions. PD is an object in a class
%   derived from the ProbDist class.
%
%   [...] = ALLFITDIST(DATA,SORTBY) returns the struct of valid distributions
%   sorted by the parameter SORTBY
%        NLogL - Negative of the log likelihood
%        BIC - Bayesian information criterion (default)
%        AIC - Akaike information criterion
%        AICc - AIC with a correction for finite sample sizes
%
%   [...] = ALLFITDIST(...,'DISCRETE') specifies it is a discrete
%   distribution and does not attempt to fit a continuous distribution
%   to the data
%
%   [...] = ALLFITDIST(...,'PDF') or (...,'CDF') plots either the PDF or CDF
%   of a subset of the fitted distribution. The distributions are plotted in
%   order of fit, according to SORTBY.
```

```
%
%   List of distributions it will try to fit
%     Continuous (default)
%       Beta
%       Birnbaum-Saunders
%       Exponential
%       Extreme value
%       Gamma
%       Generalized extreme value
%       Generalized Pareto
%       Inverse Gaussian
%       Logistic
%       Log-logistic
%       Lognormal
%       Nakagami
%       Normal
%       Rayleigh
%       Rician
%       t location-scale
%       Weibull
%
%     Discrete ('DISCRETE')
%       Binomial
%       Negative binomial
%       Poisson
%
%   Optional inputs:
%   [...] = ALLFITDIST(...,'n',N,...)
%   For the 'binomial' distribution only:
%      'n'            A positive integer specifying the N parameter (number
%                     of trials).  Not allowed for other distributions. If
%                     'n' is not given it is estimate by Method of Moments.
%                     If the estimated 'n' is negative then the maximum
%                     value of data will be used as the estimated value.
%   [...] = ALLFITDIST(...,'theta',THETA,...)
%   For the 'generalized pareto' distribution only:
%      'theta'        The value of the THETA (threshold) parameter for
%                     the generalized Pareto distribution. Not allowed for
%                     other distributions. If 'theta' is not given it is
%                     estimated by the minimum value of the data.
%
%   Note: ALLFITDIST does not handle nonparametric kernel-smoothing,
%   use FITDIST directly instead.
%
%
%   EXAMPLE 1
%     Given random data from an unknown continuous distribution, find the
%     best distribution which fits that data, and plot the PDFs to compare
%     graphically.
%        data = normrnd(5,3,1e4,1);         %Assumed from unknown
distribution
%        [D PD] = allfitdist(data,'PDF');   %Compute and plot results
%        D(1)                               %Show output from best fit
%
%   EXAMPLE 2
%     Given random data from a discrete unknown distribution, with frequency
%     data, find the best discrete distribution which would fit that data,
```

```
%     sorted by 'NLogL', and plot the PDFs to compare graphically.
%        data = nbinrnd(20,.3,1e4,1);
%        values=unique(data); freq=histc(data,values);
%        [D PD] =
allfitdist(values,'NLogL','frequency',freq,'PDF','DISCRETE');
%        PD{1}
%
%  EXAMPLE 3
%     Although the Geometric Distribution is not listed, it is a special
%     case of fitting the more general Negative Binomial Distribution. The
%     parameter 'r' should be close to 1. Show by example.
%        data=geornd(.7,1e4,1); %Random from Geometric
%        [D PD]= allfitdist(data,'PDF','DISCRETE');
%        PD{1}
%
%  EXAMPLE 4
%     Compare the resulting distributions under two different assumptions
%     of discrete data. The first, that it is known to be derived from a
%     Binomial Distribution with known 'n'. The second, that it may be
%     Binomial but 'n' is unknown and should be estimated. Note the second
%     scenario may not yield a Binomial Distribution as the best fit, if
%     'n' is estimated incorrectly. (Best to run example a couple times
%     to see effect)
%        data = binornd(10,.3,1e2,1);
%        [D1 PD1] = allfitdist(data,'n',10,'DISCRETE','PDF'); %Force binomial
%        [D2 PD2] = allfitdist(data,'DISCRETE','PDF');       %May be binomial
%        PD1{1}, PD2{1}                                     %Compare distributions
%

%    Mike Sheppard
%    Last Modified: 17-Feb-2012




%% Check Inputs
if nargin == 0
    data = 10.^((normrnd(2,10,1e4,1))/10);
    sortby='BIC';
    varargin={'CDF'};
end
if nargin==1
    sortby='BIC';
end
sortbyname={'NLogL','BIC','AIC','AICc'};
if ~any(ismember(lower(sortby),lower(sortbyname)))
    oldvar=sortby; %May be 'PDF' or 'CDF' or other commands
    if isempty(varargin)
        varargin={oldvar};
    else
        varargin=[oldvar varargin];
    end
    sortby='BIC';
end
if nargin < 2, sortby='BIC'; end
distname={'beta', 'birnbaumsaunders', 'exponential', ...
```

```
    'extreme value', 'gamma', 'generalized extreme value', ...
    'generalized pareto', 'inversegaussian', 'logistic', 'loglogistic', ...
    'lognormal', 'nakagami', 'normal', ...
    'rayleigh', 'rician', 'tlocationscale', 'weibull'};
if ~any(strcmpi(sortby,sortbyname))
    error('allfitdist:SortBy','Sorting must be either NLogL, BIC, AIC, or
AICc');
end
%Input may be mixed of numeric and strings, find only strings
vin=varargin;
strs=find(cellfun(@(vs)ischar(vs),vin));
vin(strs)=lower(vin(strs));
%Next check to see if 'PDF' or 'CDF' is listed
numplots=sum(ismember(vin(strs),{'pdf' 'cdf'}));
if numplots>=2
    error('ALLFITDIST:PlotType','Either PDF or CDF must be given');
end
if numplots==1
    plotind=true; %plot indicator
    indxpdf=ismember(vin(strs),'pdf');
    plotpdf=any(indxpdf);
    indxcdf=ismember(vin(strs),'cdf');
    vin(strs(indxpdf|indxcdf))=[]; %Delete 'PDF' and 'CDF' in vin
else
    plotind=false;
end
%Check to see if discrete
strs=find(cellfun(@(vs)ischar(vs),vin));
indxdis=ismember(vin(strs),'discrete');
discind=false;
if any(indxdis)
    discind=true;
    distname={'binomial', 'negative binomial', 'poisson'};
    vin(strs(indxdis))=[]; %Delete 'DISCRETE' in vin
end
strs=find(cellfun(@(vs)ischar(vs),vin));
n=numel(data); %Number of data points
data = data(:);
D=[];
%Check for NaN's to delete
deldatanan=isnan(data);
%Check to see if frequency is given
indxf=ismember(vin(strs),'frequency');
if any(indxf)
    freq=vin{1+strs((indxf))}; freq=freq(:);
    if numel(freq)~=numel(data)
        error('ALLFITDIST:PlotType','Matrix dimensions must agree');
    end
    delfnan=isnan(freq);
    data(deldatanan|delfnan)=[]; freq(deldatanan|delfnan)=[];
    %Save back into vin
    vin{1+strs((indxf))}=freq;
else
    data(deldatanan)=[];
end
```

```
%% Run through all distributions in FITDIST function
warning('off','all'); %Turn off all future warnings
for indx=1:length(distname)
    try
        dname=distname{indx};
        switch dname
            case 'binomial'
                PD=fitbinocase(data,vin,strs); %Special case
            case 'generalized pareto'
                PD=fitgpcase(data,vin,strs); %Special case
            otherwise
                %Built-in distribution using FITDIST
                PD = fitdist(data,dname,vin{:});
        end

        NLL=PD.NLogL; % -Log(L)
        %If NLL is non-finite number, produce error to ignore distribution
        if ~isfinite(NLL)
            error('non-finite NLL');
        end
        num=length(D)+1;
        PDs(num) = {PD}; %#ok<*AGROW>
        k=numel(PD.Params); %Number of parameters
        D(num).DistName=PD.DistName;
        D(num).NLogL=NLL;
        D(num).BIC=-2*(-NLL)+k*log(n);
        D(num).AIC=-2*(-NLL)+2*k;
        D(num).AICc=(D(num).AIC)+((2*k*(k+1))/(n-k-1));
        D(num).ParamNames=PD.ParamNames;
        D(num).ParamDescription=PD.ParamDescription;
        D(num).Params=PD.Params;
        D(num).Paramci=PD.paramci;
        D(num).ParamCov=PD.ParamCov;
        D(num).Support=PD.Support;
    catch err %#ok<NASGU>
        %Ignore distribution
    end
end
warning('on','all'); %Turn back on warnings
if numel(D)==0
    error('ALLFITDIST:NoDist','No distributions were found');
end
```

```
%% Sort distributions
indx1=1:length(D); %Identity Map
sortbyindx=find(strcmpi(sortby,sortbyname));
switch sortbyindx
```

```
    case 1
        [~,indx1]=sort([D.NLogL]);
    case 2
        [~,indx1]=sort([D.BIC]);
    case 3
        [~,indx1]=sort([D.AIC]);
    case 4
        [~,indx1]=sort([D.AICc]);
end
%Sort
D=D(indx1); PD = PDs(indx1);







%% Plot if requested
if plotind;
    plotfigs(data,D,PD,vin,strs,plotpdf,discind)
end


end






function PD=fitbinocase(data,vin,strs)
%% Special Case for Binomial
% 'n' is estimated if not given
vinbino=vin;
%Check to see if 'n' is given
indxn=any(ismember(vin(strs),'n'));
%Check to see if 'frequency' is given
indxfreq=ismember(vin(strs),'frequency');
if ~indxn
    %Use Method of Moment estimator
    %E[x]=np, V[x]=np(1-p) -> nhat=E/(1-(V/E));
    if isempty(indxfreq)||~any(indxfreq)
        %Raw data
        mnx=mean(data);
        nhat=round(mnx/(1-(var(data)/mnx)));
    else
        %Frequency data
        freq=vin{1+strs(indxfreq)};
        m1=dot(data,freq)/sum(freq);
        m2=dot(data.^2,freq)/sum(freq);
        mnx=m1; vx=m2-(m1^2);
        nhat=round(mnx/(1-(vx/mnx)));
    end
    %If nhat is negative, use maximum value of data
    if nhat<=0, nhat=max(data(:)); end
    vinbino{end+1}='n'; vinbino{end+1}=nhat;
```

```
end
PD = fitdist(data,'binomial',vinbino{:});
end
```

```
function PD=fitgpcase(data,vin,strs)
%% Special Case for Generalized Pareto
% 'theta' is estimated if not given
vingp=vin;
%Check to see if 'theta' is given
indxtheta=any(ismember(vin(strs),'theta'));
if ~indxtheta
    %Use minimum value for theta, minus small part
    thetahat=min(data(:))-10*eps;
    vingp{end+1}='theta'; vingp{end+1}=thetahat;
end
PD = fitdist(data,'generalized pareto',vingp{:});
end
```

```
function plotfigs(data,D,PD,vin,strs,plotpdf,discind)
%Plot functionality for continuous case due to Jonathan Sullivan
%Modified by author for discrete case

%Maximum number of distributions to include
%max_num_dist=Inf;  %All valid distributions
max_num_dist=4;

%Check to see if frequency is given
indxf=ismember(vin(strs),'frequency');
if any(indxf)
    freq=vin{1+strs((indxf))};
end

figure

%% Probability Density / Mass Plot
if plotpdf
    if ~discind
        %Continuous Data
        nbins = max(min(length(data)./10,100),50);
        xi = linspace(min(data),max(data),nbins);
        dx = mean(diff(xi));
        xi2 = linspace(min(data),max(data),nbins*10)';
        fi = histc(data,xi-dx);
        fi = fi./sum(fi)./dx;
        inds = 1:min([max_num_dist,numel(PD)]);
        ys = cellfun(@(PD) pdf(PD,xi2),PD(inds),'UniformOutput',0);
```

```
        ys = cat(2,ys{:});
        bar(xi,fi,'FaceColor',[160 188 254]/255,'EdgeColor','k');
        hold on;
        plot(xi2,ys,'LineWidth',1.5)
        legend(['empirical',{D(inds).DistName}],'Location','NE')
        xlabel('Value');
        ylabel('Probability Density');
        title('Probability Density Function');
        grid on
    else
        %Discrete Data
        xi2=min(data):max(data);
        %xi2=unique(x)'; %If only want observed x-values to be shown
        indxf=ismember(vin(strs),'frequency');
        if any(indxf)
            fi=zeros(size(xi2));
            fi((ismember(xi2,data)))=freq; fi=fi'./sum(fi);
        else
            fi=histc(data,xi2); fi=fi./sum(fi);
        end
        inds = 1:min([max_num_dist,numel(PD)]);
        ys = cellfun(@(PD) pdf(PD,xi2),PD(inds),'UniformOutput',0);
        ys=cat(1,ys{:})';
        bar(xi2,[fi ys]);
        legend(['empirical',{D(inds).DistName}],'Location','NE')
        xlabel('Value');
        ylabel('Probability Mass');
        title('Probability Mass Function');
        grid on
    end
else

%Cumulative Distribution
    if ~discind
        %Continuous Data
        [fi xi] = ecdf(data);
        inds = 1:min([max_num_dist,numel(PD)]);
        ys = cellfun(@(PD) cdf(PD,xi),PD(inds),'UniformOutput',0);
        ys = cat(2,ys{:});
        if max(xi)/min(xi) > 1e4; lgx = true; else lgx = false; end
        subplot(2,1,1)
        if lgx
            semilogx(xi,fi,'k',xi,ys)
        else
            plot(xi,fi,'k',xi,ys)
        end
        legend(['empirical',{D(inds).DistName}],'Location','NE')
        xlabel('Value');
        ylabel('Cumulative Probability');
        title('Cumulative Distribution Function');
        grid on
        subplot(2,1,2)
        y = 1.1*bsxfun(@minus,ys,fi);
        if lgx
            semilogx(xi,bsxfun(@minus,ys,fi))
        else
```

```
            plot(xi,bsxfun(@minus,ys,fi))
        end
        ybnds = max(abs(y(:)));
        ax = axis;
        axis([ax(1:2) -ybnds ybnds]);
        legend({D(inds).DistName},'Location','NE')
        xlabel('Value');
        ylabel('Error');
        title('CDF Error');
        grid on
    else
        %Discrete Data
        indxf=ismember(vin(strs),'frequency');
        if any(indxf)
            [fi xi] = ecdf(data,'frequency',freq);
        else
            [fi xi] = ecdf(data);
        end
        %Check unique xi, combine fi
        [xi,ign,indx]=unique(xi); %#ok<ASGLU>
        fi=accumarray(indx,fi);
        inds = 1:min([max_num_dist,numel(PD)]);
        ys = cellfun(@(PD) cdf(PD,xi),PD(inds),'UniformOutput',0);
        ys=cat(2,ys{:});
        subplot(2,1,1)
        stairs(xi,[fi ys]);
        legend(['empirical',{D(inds).DistName}],'Location','NE')
        xlabel('Value');
        ylabel('Cumulative Probability');
        title('Cumulative Distribution Function');
        grid on
        subplot(2,1,2)
        y = 1.1*bsxfun(@minus,ys,fi);
        stairs(xi,bsxfun(@minus,ys,fi))
        ybnds = max(abs(y(:)));
        ax = axis;
        axis([ax(1:2) -ybnds ybnds]);
        legend({D(inds).DistName},'Location','NE')
        xlabel('Value');
        ylabel('Error');
        title('CDF Error');
        grid on
    end
end

end
```

B.3.3 Correlations

The code that check the correlation and plots the scatter plots and the autocorrelation.

```
i1=1;
machines={'[redacted]','[redacted]','[redacted]','[redacted]','[redacted]','[redacted]','[redacted]', [redacted]
[redacted] ,'[redacted]','[redacted]','[redacted]','[redacted]','[redacted]','[redacted]','[redacted]','[redacted]'};
%
for i=machines



filename=strcat('C:\Users\spiros\Desktop\matlabthesis\equipment\',i(1),'.xlsm
');
   filename1=filename{1};
   nof=xlsread(filename1,'data','V:V');

   UP=xlsread(filename1,'data',['A2:A',num2str(nof+1)]);
   PT=xlsread(filename1,'data',['B2:B',num2str(nof+1)]);
   DT=xlsread(filename1,'data',['C2:C',num2str(nof+1)]);
   MT=xlsread(filename1,'data',['D2:D',num2str(nof+1)]);
   TTR=xlsread(filename1,'data',['F2:F',num2str(nof+1)]);
   NM=xlsread(filename1,'data',['G2:G',num2str(nof+1)]);
   TT=UP+DT;


finame=strcat('C:\Users\spiros\Desktop\matlabthesis\correlations\',i{1},'PTvs
MT','.jpg');
   scatter(MT,PT)
   title(strjoin({'Maintenance Time VS Processing Time for', i{1}}))
   xlabel('Maintenance Time')
   ylabel('Processing Time')
   [RHO,PVAL] = corr(PT,MT,'Type','Spearman');
   x = input('wait')
   saveas(gcf,finame)
   A={i{1},'PTvsMT',num2str(RHO),num2str(PVAL)};
   i1=i1+1;

xlswrite('C:\Users\spiros\Desktop\matlabthesis\results.xlsx',A,'correlations'
,['A',num2str(i1),':k',num2str(i1)])


finame=strcat('C:\Users\spiros\Desktop\matlabthesis\correlations\',i{1},'PTvs
NM','.jpg');
   scatter(NM,PT)
   [RHO,PVAL] = corr(PT,NM,'Type','Spearman');
   title(strjoin({'# of Maintenace VS Processing Time for', i{1}}))
   xlabel('# of Maintenace')
   ylabel('Processing Time ')
   %x = input('wait')
   %saveas(gcf,finame)
   A={i{1},'PTvsNM',num2str(RHO),num2str(PVAL)};
```

```
   i1=i1+1;

xlswrite('C:\Users\spiros\Desktop\matlabthesis\results.xlsx',A,'correlations'
,['A',num2str(i1),':k',num2str(i1)])


finame=strcat('C:\Users\spiros\Desktop\matlabthesis\correlations\',i{1},'PTvs
TTR','.jpg');
   scatter(PT(1:(nof-1)),TTR(2:nof))
   title(strjoin({'Processing Time VS Time to Restore for', i{1}}))
   xlabel('Processing Time')
   ylabel('Time to Restore')
   [RHO,PVAL] = corr(PT(1:(nof-1)),TTR(2:nof),'Type','Spearman');
   %x = input('wait')
   %saveas(gcf,finame)
   A={i{1},'PTvsTTR',num2str(RHO),num2str(PVAL)};
   i1=i1+1;

xlswrite('C:\Users\spiros\Desktop\matlabthesis\results.xlsx',A,'correlations'
,['A',num2str(i1),':k',num2str(i1)])


finame=strcat('C:\Users\spiros\Desktop\matlabthesis\correlations\',i{1},'MTvs
TTR','.jpg');
   scatter(MT(1:(nof-1)),TTR(2:nof))
   title(strjoin({'Maintenance Time VS Time to Restore for', i{1}}))
   xlabel('Maintenance Time')
   ylabel('Time to Restore')
   [RHO,PVAL] = corr(MT(1:(nof-1)),TTR(2:nof),'Type','Spearman');
   %x = input('wait')
   %saveas(gcf,finame)
   A={i{1},'MTvsTTR',num2str(RHO),num2str(PVAL)};
   i1=i1+1;

xlswrite('C:\Users\spiros\Desktop\matlabthesis\results.xlsx',A,'correlations'
,['A',num2str(i1),':k',num2str(i1)])


finame=strcat('C:\Users\spiros\Desktop\matlabthesis\correlations\',i{1},'Auto
corr','.jpg');
   autocorr(TT)
   title(strjoin({'Autocorrelation of time between failures for', i{1}}))
   %saveas(gcf,finame)


end
```